

XenApp 和 XenDesktop 7.6

Nov 03, 2016

新增功能

[长期服务版本 \(Long Term Service Release, LTSR\)](#)

[Feature Pack 3 : 安装和升级](#)

[Feature Pack 3 : 已修复的问题](#)

[已知问题](#)

[本版本中不提供的功能](#)

系统要求

技术概述

[概念和组件](#)

[Active Directory](#)

[容错](#)

[交付方法](#)

[参考体系结构](#)

[设计指南](#)

[实施指南](#)

新部署

[准备安装](#)

[准备虚拟化环境 : VMware](#)

[准备虚拟化环境 : Microsoft System Center Virtual Machine Manager](#)

[准备使用 Microsoft System Center Configuration Manager](#)

[使用图形界面进行安装](#)

[使用命令行安装](#)

[创建站点](#)

[使用脚本安装或删除 Virtual Delivery Agent](#)

[计算机目录](#)

[交付组](#)

[XenApp 发布的应用程序和桌面](#)

[VM 托管应用程序](#)

[VDI 桌面](#)

[Remote PC Access](#)

[App-V](#)

[本地应用程序访问和 URL 重定向](#)

[服务器 VDI](#)

[删除组件](#)

升级和迁移

[升级部署](#)

[迁移 XenApp 6.x](#)

[迁移 XenDesktop 4](#)

安全性

[Citrix XenApp 和 XenDesktop 安全性入门](#)

[安全最佳实践和注意事项](#)

[委派管理](#)

[智能卡](#)

[SSL](#)

策略

[使用策略](#)

[策略模板](#)

[创建策略](#)

[对策略进行比较、设定优先级、建模和故障排除](#)

[默认策略设置](#)

[策略设置参考](#)

打印

[打印配置示例](#)

[最佳做法、安全注意事项和默认操作](#)

[打印策略和首选项](#)

[置备打印机](#)

[维护打印环境](#)

许可

连接和资源

连接租用

虚拟 IP 和虚拟环回

辅助数据库位置

Delivery Controller 环境

添加、删除或移动 Controller 或者移动 VDA

基于 Active Directory OU 的控制器发现

会话管理

在 Studio 中使用搜索

IPv4/IPv6 支持

客户端文件夹重定向

Personal vDisk

安装和升级

配置与管理

工具

显示、消息和故障排除

用户配置文件

HDX

Thinwire 兼容模式

HDX 3D Pro

Flash 重定向

Host to client redirection

适用于 Windows 桌面操作系统的 GPU 加速

Windows Server 操作系统 GPU 加速

OpenGL Software Accelerator

音频功能

网络流量优先级

[USB 和客户端设备注意事项](#)

[监视](#)

[Director](#)

[Session Recording](#)

[Personal vDisk](#)

[配置日志记录](#)

[Monitor Service OData API](#)

[SDK](#)

[了解 XenDesktop 管理模式](#)

[SDK 入门](#)

[PowerShell cmdlet 帮助](#)

[FIPS 部署示例](#)

[第三方声明](#)

[适用于 XenApp 和 XenDesktop 的 Citrix SCOM 管理包](#)

[适用于许可证服务器的 Citrix SCOM 管理包](#)

新增功能

Nov 02, 2016

In this article:

[Long Term Service Release \(LTSR\)](#)

[Feature Pack 3](#)

[Feature Pack 2](#)

[Feature Pack 1](#)

[XenApp and XenDesktop 7.6](#)

Feature Pack 3

The Feature Pack 3 release includes the following new enhanced features.

For install and upgrade instructions, see [Install and upgrade Feature Pack 3 components](#).

Fixed Issues

The updated VDA standalone package adds:

- **Support for Windows 10** Enterprise Edition, in the Standard VDA for Windows Desktop OS.

The updated Universal Print Server package adds:

- **Support for Windows Server 2012 R2.**

The following new and enhanced HDX technologies features are included in the Feature Pack 3 VDA and managed using the updated Group Policy Management package:

- **Enhancements to Thinwire Compatibility Mode.** This mode uses an updated encoder that improves the display performance of simple graphical interfaces, for example in MS Excel and MS Word, without requiring the use of a resource-intensive video codec. For more information, see [Thinwire Compatibility Mode](#).
- **Support for signature devices and drawing tablets.** Administrators can set a policy to enable devices for client-side polling and set optimization modes as part of that policy. This means that users can work effectively in wide area networks with devices such as Wacom signature pads and tablets within XenApp and XenDesktop sessions. For more information, see [USB devices policy settings](#).
- **Video fallback prevention policy.** Enables administrators to have more control over server-side and client-side fetching and rendering of multimedia content. For example, if a user's device cannot play a video locally, a new setting

allows the administrator to control whether the video is automatically rendered on the server, or prevented and a message displayed to the user. For more information, see [Multimedia policy settings](#) and [Flash Redirection policy settings](#).

- **Minimum version checking for Flash redirection.** For client devices accessing VDAs using Receiver for Windows and Receiver for Linux. This gives administrators greater control over specifying minimum versions required for Flash redirection. For more information on how to specify minimum versions, see [Flash Redirection](#).
- **Smart card virtual channel improvements.** Includes support for smart card reader plug-and-play with Receiver for Windows 4.3, virtual smart cards (Microsoft Windows Trusted Platform Module-based virtual smart cards), and performance improvements. For more information on deploying smart cards, see [Smart cards](#), and [Virtual Smart Card Overview](#) in the Microsoft Windows TechCenter.
- **Framehawk virtual display channel integration with the VDA.** The Framehawk virtual display channel is integrated into the standalone VDA package. With Framehawk integrated into HDX technology, Citrix provides a smooth and intuitive user experience in conditions where remote workspace users previously experienced poor interactivity, such as on wireless connections with high packet loss or congestion. For more information, see the [HDX Framehawk virtual channel administrator guide](#).
- **Full-screen app support for HDX 3D Pro VDI.** The Virtual Delivery Agent (VDA) for HDX 3D Pro supports full-screen apps including 3D and gaming apps within single monitor ICA sessions.
- **Custom hotkey configuration for lossless compression.** With HDX 3D Pro, the administrator can configure a registry setting to set a custom keyboard shortcut to override the default (ALT+SHIFT+1). Users can instantly switch between lossy and visually lossless compression, or between lossy and lossless compression using a custom hotkey. For more information, see [GPU acceleration for Windows Desktop OS](#).
- **Call Home.** In this version of Call Home, administrators can run a PowerShell cmdlet (**Start-TelemetryUpload**) to extract VDA diagnostic information. You can use the cmdlet to either create a local ZIP file, which you can email to your support representative, or you can use credential information combined with a Support Service Request (SR) number to automatically upload the information to Citrix Insights Services. This enables administrators and support representatives to identify and troubleshoot VDA-related issues. The PowerShell cmdlet is supplied with the VDA installer in FP3.
- **Enhanced HDX printer support and configuration.** Includes always-on logging for the print server and printing subsystem on the VDA, and the option to join the Citrix Customer Experience Improvement Program (CEIP). For more information, see [Install using the command line](#) and [Best practices, security considerations, and default operations](#).
- **Updated Policy templates.** This feature pack delivers new policy templates for High Server Scalability and Optimized for WAN, and replaces the High Definition User Experience template with the Very High Definition template. For more information, see [Policy templates](#).
- **32-bit cursor support.** Delivers an improved user experience in 3D applications which rely on custom cursors.
- **Desktop Composition Redirection (DCR).** DCR is disabled by default.

The updated StoreFront package adds:

- **Support for Transport Layer Security (TLS) 1.0 and 1.1.** Secure Sockets Layer (SSL) 3.0 is **not** supported and Citrix strongly recommends that you do not use it.

See [StoreFront](#) documentation for more information.

You can create Linux virtual desktops based on SUSE and Red Hat distributions. Prepare your Linux virtual machines, install the new [Linux VDA software](#) on them, configure your Delivery Controller, and then use Studio to make the desktops available to users. For more information, see the following documents:

 [Installation guide for SUSE](#)

 [Installation guide for Red Hat](#)

Feature Pack 2

The Feature Pack 2 release includes the following new and enhanced features.

StoreFront includes the following new features and enhancements.

Updated StoreFront management console

Support for the unified Receiver experience. StoreFront 3.0 delivers centralized customization and branding of your end users' applications and desktop selection experience to Receiver users. Your company can customize the interface to reflect your logo, colors, and so forth. Some of the new Receiver and StoreFront features:

- **Simplified application organization.** Familiar and consistent application selection experience for use.
 - Featured app groups. Administrator configured applications that are logically grouped and advertised to users in the applications selection experience.
 - Folder view of applications. Return of Web Interface functionality in StoreFront.
 - Favorites. Easily add or remove applications to Favorites for quick access.
- **Server-managed user experience.** Application selection experience is server configured.
 - Change once, deploy everywhere. User application selection experience changes are made by the administrator on StoreFront server.
 - Consistent application selection experience for users. Any device running Receiver receives its application selection experience from the server as HTML5, providing for consistency as users move across smartphones, tablets, and desktop form factors.
 - Decouple the application selection experience from Receiver capabilities. Client upgrade without end user training or user experience impact; Support legacy user experience without legacy client cost.
- **Fit for purpose user experience customization options.** You select the appropriate level of customization.
 - Updated out-of-the-box user experience. Modernized end user application selection look-and-feel.
 - Server-configured branding customizations. You brand with corporate logos, color, and featured application groups.
 - Deeper customization and branding options. CSS and script APIs available to enable deeper levels of customization.
- **Use the StoreFront management console to do the following Receiver related tasks:**
 - Set Receiver for Web as the default for the store.
 - Create a Receiver for Web website.
 - Change website type.
 - Customize the website appearance.

See the [Manage a Citrix Receiver for Web site](#) articles.

- **Featured app groups management.** App groups are groups of applications that are related or fit in a specific category making them more easily discoverable. With Studio and the StoreFront management console you can define app groups using keywords, application names, or categories.

See the [Create and manage featured apps](#) article.

XML service-based authentication

If you configure all of them, the event that occurs first will cause the unused session to end. Use PowerShell cmdlets to enable and disable this feature. For more information, see [XML service-based authentication](#).

HDX RealTime Optimization Pack includes the following new features and enhancements.

- Support for the Microsoft Skype for Business client in Lync UI mode, the Microsoft Lync 2013 client, and the Microsoft Lync 2010 client.
- Support for the Lync Server 2013 Autodiscover Service.
- Call Park and Call Pick Up in Lync 2010 client.
- Call forwarding and simultaneous ringing controls for Lync 2010 client
- Support for Mac.
- Kerberos authentication.
- Plus (+) symbol in the dial pad.
- Support for the Microsoft Windows 10 technical preview.

For information, see [HDX RealTime Optimization Pack 1.8 for Microsoft Lync](#).

Session Recording includes the following new features and enhancements.

- You can specify the connection credentials to the database when installing the Session Recording Database component.
- You can test the connectivity of database during the installation of the Session Recording Database and Session Recording Server components and test the connectivity of the Session Recording Server during the installation of Session Recording Agent component.
- Microsoft Shared Management Objects is no longer requirement for Session Recording Database installation.
- Citrix Experience Improvement Program (CEIP) is integrated in Session Recording. For more information, see [About the Citrix Customer Experience Improvement Program](#).

For more information, see Session [Recording - for XenApp 7.6 FP1, FP2, and LTSR](#).

You can now create Linux virtual desktops based on SUSE and Red Hat distributions. Prepare your Linux virtual machines, install the new [Linux VDA software](#) on them, configure your Delivery Controller, and then use Studio to make the desktops available to users. For more information, see the latest documents in the Feature Pack 3 section above.

Framehawk is a new ICA virtual channel that extends Citrix HDX technologies, a set of capabilities that work together to deliver a high-definition user experience of virtual desktops and applications.

The Framehawk virtual channel optimizes the delivery of virtual desktops and applications to users on broadband wireless connections. It is ideal for mobile devices under lossy network connections. With Framehawk integrated into the industry-leading HDX technology, Citrix provides a smooth and intuitive user experience in conditions where remote workspace users previously experienced poor interactivity, such as on wireless connections with high packet loss or congestion.

For more information, see the following document which contains information for both XenApp and XenDesktop 7.6 Feature Pack 3 and Feature Pack 2.

 [Framehawk administrator guide](#)

This version of Director supports the Framehawk virtual channel and contains fixes for the following issues in that were present in Version 7.6.200. For more information on installing this component, see the [Director documentation](#).

Fixed issues

Feature Pack 1

The Feature Pack 1 release includes the following new and enhanced features.

Session Recording allows you to record the on-screen activity, over any type of connection, from any server running XenApp subject to corporate policy and regulatory compliance. Session Recording records, catalogs, and archives sessions for retrieval and playback. For more information, see [Session Recording - for XenApp 7.6 FP1, FP2, and LTSR](#).

With Director 7.6.200, you can enable and disable the Session Recording policy from the User Details and Machine Details views. Use the Director console to create and activate Session Recording policies.

For more details about the integration of Session Recording and Director, see <http://support.citrix.com/article/CTX142260>. For more information about Session Recording, see [Session Recording - for XenApp 7.6 FP1, FP2, and LTSR](#).

Citrix Licensing Customer Experience Improvement Program (CEIP) and Call Home — Voluntary data collection programs in which Citrix products gather anonymous or identified configuration, performance, error, and usage data from your deployment and automatically send the data to Citrix. For more information, see [Citrix Licensing Customer Experience Improvement Program \(CEIP\) and Call Home](#). Not supported with License Server VPX.

The Citrix Customer Experience Improvement Program (CEIP) gives you the opportunity to contribute to the design and development of Citrix products. When you enroll in the program, Citrix collects anonymous information about your deployment, which is used to improve product quality, reliability, and performance.

For more information, see [About the Citrix Customer Experience Improvement Program](#).

Citrix HDX RealTime Optimization Pack 1.7 for Microsoft Lync provides a highly scalable solution for delivering real-time audio-video conferencing and VoIP enterprise telephony through Microsoft Lync in XenDesktop and XenApp environments to users on Linux and Windows devices. The Optimization Pack leverages your existing Microsoft Lync infrastructure and interoperates with other Microsoft Lync endpoints running natively on devices.

For information, see [HDX RealTime Optimization Pack 1.7 for Microsoft Lync](#).

XenApp 7.6 and XenDesktop 7.6

This product release includes the following new and enhanced features.

The session prelaunch and session linger features help users quickly access applications by starting sessions before they are requested (session prelaunch) and keeping application sessions active after a user closes all applications (session linger). These features are supported for Server OS machines only.

By default, session prelaunch and session linger are not used. A session starts (launches) when a user starts an application and remains active until the last open application in the session closes. You can enable the features for all users in a Delivery Group or only for specified users.

There are several ways to specify how long an unused session remains active if the user does not start an application: a configured timeout and two server load thresholds. You can configure all of them; the event that occurs first will cause the unused session to end.

For more information, see [Configure session prelaunch and session linger](#).

When creating or editing Delivery Groups containing Server OS machines, you can now allow users to access applications and desktops without presenting credentials to StoreFront or Citrix Receiver. For example, when users access applications through kiosks, the application may require credentials, but the Citrix access portal and tools do not.

When you configure the Delivery Group, you can grant access to authenticated users, unauthenticated users, or both. When you grant access to unauthenticated (anonymous) users, you must provide an unauthenticated StoreFront store.

For more information, see [Users](#).

To ensure that the Site database is always available, Citrix recommends starting with a fault-tolerant SQL Server deployment by following high availability best practices from Microsoft. However, network issues and interruptions may prevent Delivery Controllers from accessing the database, resulting in users not being able to connect to their applications or desktop.

The connection leasing feature supplements the SQL Server high availability best practices by enabling users to connect and reconnect to their most recently used applications and desktops, even when the Site database is not available.

Although users may have a large number of published resources available, they often use only a few of them regularly.

When you enable connection leasing, each Controller caches user connections to those recently used applications and desktops during normal operations (when the database is available). If the database becomes unavailable, the Controller enters leased connection mode and "replays" the cached operations when a user attempts to connect or reconnect to a recently used application or desktop from StoreFront.

For more information, see [Connection leasing](#).

You can organize applications in folders, which makes it easier to administer large numbers of applications in Studio.

By default, applications in a Delivery Group appear in a single folder. From the Delivery Group display in Studio, you can create additional folders and move applications into them. Moving, nesting, and renaming folders are easy drag-and-drop operations; you can also use Actions menu items. Additionally, you can specify folder destinations and create new folders when you add applications to a Delivery Group.

For more information, see [Manage application folders](#).

The XenApp 6.5 migration process helps you more efficiently and quickly transition from a XenApp 6.5 farm to a Site running XenApp 7.6 (or a later supported release). This is helpful in deployments that contain large numbers of applications and Citrix group policies, lowering the risk of inadvertently introducing errors when manually moving applications and Citrix group policies to the new XenApp Site.

After you install the XenApp 7.6 core components and create a Site, the migration process follows this sequence:

- Run the XenApp 7.6 installer on each XenApp 6.5 worker, which automatically upgrades it to a new Virtual Delivery Agent for Windows Server OS for use in the new Site.
- Run PowerShell export cmdlets on a XenApp 6.5 controller, which exports application and Citrix policy settings to XML files.
- Edit the XML files, if desired, to refine what you want to import to the new Site. By tailoring the files, you can import policy and application settings into your XenApp 7.6 Site in stages: some now and others later.
- Run PowerShell import cmdlets on the new XenApp 7.6 Controller, which import settings from the XML files to the new XenApp Site.
- Reconfigure the new Site as needed, and then test it.

For more information, see [Migrate XenApp 6.x](#).

The Citrix Customer Experience Improvement Program (CEIP) gives you the opportunity to contribute to the design and development of Citrix products. When you enroll in the program, Citrix collects anonymous information about your deployment, which is used to improve product quality, reliability, and performance.

It's easy to enroll in the program after you create or upgrade a Site. You can also opt in or out of the program at any time by selecting Configuration in the Studio navigation pane and following the instructions.

For more information, see [About the Citrix Customer Experience Improvement Program](#).

To improve performance, you can now specify the maximum number of simultaneous actions, simultaneous Personal

Storage inventory updates, and actions per minute that can occur on a host connection.

For more information, see [Edit a connection](#).

Studio displays more detailed status and error reporting when updating PvD images, and displays comprehensive licensing alerts when you are in the licensing node.

You can enable Secure Sockets Layer (SSL/TLS) connections between users and VDAs by configuring SSL/TLS on the machines where the VDAs are installed and in the Delivery Groups that contain the VDAs.

For more information, see [SSL](#).

For published applications, you can enable and use the Microsoft virtual IP feature in machines running Windows Server 2008 R2 and Windows Server 2012 R2. Additionally, you can add new Citrix policy settings to manage virtual loopback. A preferred loopback option is also available.

For more information, see [Virtual IP and virtual loopback](#).

You can restrict or enable sharing of data from the host clipboard to the client or from the client clipboard to the user session. In each case, when sharing is restricted, you can specify data formats that can be shared.

For more information, see [ICA policy settings](#).

You can now prevent a local user from disconnecting a remote session without the permission of the remote user.

When disconnecting a remote session, moving the mouse or pressing a keyboard key wakes the local monitor. (In previous releases, pressing CTRL+ALT+DEL twice presented the logon screen.)

Icon locations are now preserved when connecting from a lower resolution device and then returning to a larger resolution device.

This release provides support for Generic USB Redirection for specialty USB devices for which there is no optimized virtual channel. This functionality redirects arbitrary USB devices from client machines to virtual desktops; with this feature, end users have the ability to interact with a wide selection of generic USB devices in the desktop session as if the devices were physically attached.

With Generic USB Redirection:

- users do not need to install device drivers on the user device
- USB client drivers are installed on the VDA machine

This feature requires Windows Server 2012 R2, and functions with existing Windows Receiver versions for published desktop

sessions hosted on RDS hosts in single-hop scenarios. Using this feature, USB client drivers are installed on the host, so these drivers must be compatible with RDSH for Windows 2012 R2 platforms.

Director 7.6.100 includes the following new and enhanced features. For download, installation, and upgrade information, see <http://support.citrix.com/article/CTX200330>.

- **Virtual machine usage.** Provides administrators with the real-time view of their VM usage so they can quickly assess their site's capacity needs. VM usage is categorized by Desktop OS availability and Server OS availability.
 - **Desktop OS availability.** Displays the current state of Desktop OS machines (VDIs) by availability for the entire site or specific Delivery Group.
 - **Server OS availability.** Displays the current state of Server OS machines by availability for the entire site or specific Delivery Group.
- **Export improvements.** Enhanced to provide the option of exporting the trends reports in CSV, PDF, and Excel formats.
- **Zoom-in drilldown enhancements.** Drilldown capabilities were added to Director 7.6. This enhanced feature lets administrators navigate through trend charts by zooming in on a time period (clicking on a data point in the graph) and drilling down to see the details associated with the trend. The administrators can now better understand the details of who or what has been affected by the trends being displayed.

Licensing alerts making you aware of issues that may impact user connections. Director also displays a recommended action to correct the condition. Some of the conditions displayed in Director are:

- All licenses have expired.
- Licenses are about to expire.
- Citrix license grace period has expired.
- The Supplemental Grace Period is active, and all installed licenses are currently in use.

View hosted applications usage. You can select the Delivery Group and time period to view a graph displaying peak concurrent usage and a table displaying application-based usage. From the Application Based Usage table, you can choose a specific application to see details and a list of users who are using or have used the application.

Monitor hotfixes. You can view the hotfixes installed on a specific machine VDA (physical or VM) using the User Details or Machine Details view.

The filtering feature has been expanded. Filter data is clickable and leads to User Details, Machine Details, Endpoint Details, and Anonymous Sessions.

Director is compatible with XenApp 6.5. You can use Director to monitor your XenApp 6.5 deployments.

For information, see [Director](#).

Citrix AppDNA accelerates the migration and transformation of desktop and web applications for new environments through rapid analysis, automated application remediation and packaging, and daily application management. AppDNA 7.6 includes a new Build Assessment solution that tests whether applications will work on additional builds of the same OS family. AppDNA 7.6 analysis enhancements now indicate whether required applications, application frameworks, and files are present, whether enabled GPOs will cause issues, and whether web applications are compatible with Citrix WorxWeb.

For information, see [AppDNA 7.6](#).

Simplified store configuration in the administration console. The updated StoreFront console simplifies the StoreFront configuration for the following features:

- User subscriptions
- Set session timeout for Receiver for Web
- Show domains list in logon page

Receiver for Web My Apps Folder View. This new view displays the applications in a folder hierarchy and includes a breadcrumb path for unauthenticated and mandatory stores. This folder view can help your users move from Web Interface to Receiver for Web.

Kerberos constrained delegation for XenApp 6.5. StoreFront with Kerberos constrained delegation enables pass-through authentication, eliminating the need for the client and device to run Windows with Receiver.

Single Fully Qualified Domain Name (FQDN) access. With this feature, you can provide access to resources internally and externally using a single FQDN.

XenApp Services Support smart card authentication. The StoreFront server authenticates using smart cards to XenApp Services Support sites and does not require specific versions of Receiver and operating systems.

Receiver for Android, iOS, and Linux smart card authentication. New versions of Receiver support local and remote use of smart cards for accessing apps and desktops.

Extensible authentication. Support for extensible authentication provides a single customization point for extension of StoreFront's form-based authentication. Worx Home and Receiver for Web use it to authenticate with XenMobile and XenApp and XenDesktop for both internal (direct) and external (using NetScaler Gateway) access scenarios.

For information, see [StoreFront 2.6](#).

Citrix Connector 7.5 provides a bridge between Microsoft System Center Configuration Manager and XenApp or XenDesktop, enabling you to extend the use of Configuration Manager to your Citrix environments. Citrix Connector 7.5 support now includes the Platinum editions of XenApp 7.6 and XenDesktop 7.6.

For information, see [Citrix Connector 7.5 for System Center Configuration Manager 2012](#).

Receiver for Chrome enables users to access virtual desktops and hosted applications from devices running the Google Chrome operating system. Users access these resources through Receiver for Chrome, and their desktops and applications are displayed in a single window.

Receiver for HTML5 is hosted on StoreFront servers and enables users to access virtual desktops and hosted applications from a web browser. Users can access desktops and applications within their web browsers without needing to install Citrix Receiver on their devices.

In this release, both these Receivers include the ability to convert documents to PDF from hosted applications or applications running on virtual desktops and view them on a local device or print to a locally attached printer; enhanced

clipboard operations; end-user experience metrics; and the ability to track licence usage for hosted applications.

For information, see [Receiver for Chrome 1.4](#) and [Receiver for HTML5 1.4](#).

HDX Real-Time Optimization Pack 1.5 for Microsoft Lync supports Lync-certified USB phones, mixed Lync 2010 client and Lync Server 2013 configuration, and asynchronous upgrades.

For information, see [HDX Real-Time Optimization Pack 1.5 for Microsoft Lync](#).

长期服务版本 (Long Term Service Release, LTSR)

Jan 04, 2017
发布日期：2016 年 1 月 11 日

安装和升级 LTSR 组件

要满足 XenApp 和 XenDesktop 7.6 长期服务版本 (LTSR) 的要求，必须升级属于 LTSR 的一部分以及 LTSR 版本部署的一部分的 XenApp 和 XenDesktop 7.6 的组件。例如：如果 Provisioning Services 属于您的部署的一部分，则必须将 Provisioning Services 组件升级到其 LTSR 版本。如果 Provisioning Services 不属于您的部署的一部分，则不需要安装或升级该组件。

根据 LTSR 条款，为使您的部署有资格享有各项优势，必须升级到 LTSR 版本。

此外，Citrix 还建议您使用特定版本的 Citrix Receiver 及其他组件。升级到这些组件的当前版本可确保进一步简化维护过程以及确保您的部署中最新修复的可用性，但这并非是 LTSR 合规性的必需条件。

有用链接：

- [下载 LTSR \(XenApp\)](#)
- [下载 LTSR \(XenDesktop\)](#)
- [XenApp 和 XenDesktop 服务选项](#)
- [LTSR 常见问题解答 \(FAQ\)](#)
- [产品生命周期日期](#)
- [Receiver for Windows 的 LTSR 计划](#)

注意

下面是 LTSR 基础版本特定的信息。有关 CU1 或 CU2 的此类信息，请参阅各自的文档。

虽然 LTSR 合规性不要求在您的部署中安装以下组件，但您必须将部署中的每个组件都升级到下文指示的版本。

VDA for Desktop OS	7.6.300	适用于 Windows 10 的特殊规则。请 参阅 兼容组件和平台 。
VDA for Server OS	7.6.300	

Delivery Controller	7.6 Update 3	
Citrix Studio	7.6 Update 3	
Citrix Director	7.6.300	
组策略管理体验	7.6.300 (2.5)	
StoreFront	3.0.1	
Provisioning Services	7.6 Update 1	适用于 Windows 10 的特殊规则。请参阅 兼容组件和平台 。
通用打印服务器	7.6.300	仅限 Windows 2008 R2 SP1 Windows 2012 支持 Windows 2012 R2
Session Recording	7.6.100	仅限 Platinum Edition

建议您在 7.6 LTSR 环境中使用以下组件。这些组件无权享有 LTSR 的优势（扩展的生命周期以及仅用于修复的累积更新）。Citrix 可能会要求您在 7.6 LTSR 环境中升级到这些组件的较新版本。

Windows 10 注意事项：可以通过当前版本路径获取对 Windows 10 的常规支持。Windows 10 无法享有 7.6 LTSR 的所有优势。对于包括 Windows 10 计算机的部署，Citrix 建议您使用 VDA for Desktop OS 7.9 和 Provisioning Services 7.9。

有关详细信息，请参阅 [Adding Windows 10 Compatibility to XenApp and XenDesktop 7.6 LTSR](#)（向 XenApp 和 XenDesktop 7.6 LTSR 添加 Windows 10 兼容性）和 [XenApp and XenDesktop Servicing Options \(LTSR\) FAQ](#)（XenApp 和 XenDesktop 服务选项 (LTSR) 常见问题解答）。

Profile Management	5.4
AppDNA	7.6.5
许可证服务器	11.12.1
HDX RealTime Optimization Pack	2.0
Windows 10	VDA : 版本 7.9

为简化维护过程以及确保实现最佳性能，Citrix 建议您在最新版本的 Citrix Receiver 可用时随时升级到相应版本。可以从 <https://www.citrix.com/downloads/citrix-receiver.html> 下载最新版本。为方便起见，请考虑订阅 [Citrix Receiver RSS 源](#) 以在新版本的 Citrix Receiver 可用时接收通知。

请注意，Citrix Receiver 无法享有 XenApp 和 XenDesktop LTSR 的优势（扩展的生命周期和仅用于修复的累积更新）。Citrix 可能会要求您在 7.6 LTSR 环境中升级到 Citrix Receiver 的较新版本。如果使用的是 Citrix Receiver for Windows，Citrix 已公布特殊的 LTSR 计划。可以从 [Citrix Receiver 的生命周期里程碑](#) 页面获取有关该计划的详细信息。

特别需要指出的是，LTSR 支持以下版本的 Citrix Receiver 以及之后的所有版本：

Citrix Receiver for Windows	4.4 或更高版本
Citrix Receiver for Linux	13.2.1 或更高版本
Citrix Receiver for Mac	12.1 或更高版本
Citrix Receiver for Chrome	1.8 或更高版本
Citrix Receiver for HTML5	1.8 或更高版本
Citrix Receiver for iOS	6.1.1 或更高版本
Citrix Receiver for Android	3.8 或更高版本

以下功能、组件和平台无法享有 LTSR 的生命周期里程碑和优势。需要特别指出的是，累积更新和扩展生命周期优势被排除在外。可以通过常规的最新版本获取扩展功能和组件的更新。

本地应用程序访问
Framehawk

Linux VDA
Personal vDisk

Windows 2008 32 位（面向通用打印服务器）

* Citrix 保留根据第三方供应商的生命周期里程碑更新平台支持的权利。

可以直接从 XenApp 和 XenDesktop 7.6 升级到 LTSR 以及从三个 7.6 Feature Pack 之一进行升级。

下载位置：

- [下载 LTSR \(XenApp\)](#)
- [下载 LTSR \(XenDesktop\)](#)

好消息：多个组件的 LTSR 版本已经发布了一段时间，主要作为 Feature Pack 3 的一部分提供。这意味着，如果您已将部署升级到 Feature Pack 3，多个组件已符合 LTSR。在这些情况下，您不需要进一步执行任何操作。请注意，本版本最初是在下文列出的各种组件的相应章节中发布的，以确认您是否需要越过 Feature Pack 3 进行升级。

升级 Controller 之前的注意事项：升级到 Controller 的 LTSR 版本会修改您的站点数据存储的一个或多个 DbSchema。这些修改具有永久性和不可逆性，即，不能自动还原这些修改。因此，升级 Controller 之前，请务必阅读并理解有关升级到 Controller 的 LTSR 版本的相关章节。

Virtual Delivery Agent (VDA) for Desktop OS 7.6.300

LTSR 版本：VDA for Desktop OS 7.6.300

本版本最初是在 2015 年 9 月 30 日作为 Feature Pack 3 的一部分发布的 (VDAWorkstationSetup_7.6.300.exe)

[系统要求](#)

[已修复的问题](#)

安装/升级

在要安装 VDA 的计算机上下载并运行 VDAWorkstationSetup_7.6.300.exe。请使用图形界面或命令行。

有关详细信息，请参阅[使用独立的软件包安装 VDA](#)。

Virtual Delivery Agent (VDA) for Server OS 7.6.300

LTSR 版本：VDA for Server OS 7.6.300

本版本最初是在 2015 年 9 月 30 日作为 Feature Pack 3 的一部分发布的 (VDAServerSetup_7.6.300.exe)

系统要求

已修复的问题

安装/升级

在要安装 VDA 的计算机上下载并运行 VDAServerSetup_7.6.300.exe。请使用图形界面或命令行。

VDA for Windows Server OS 安装会自动部署 Microsoft Visual C++ 2013 Runtime (32 位和 64 位) 以及 2008 和 2010 Runtime (32 位和 64 位)。不再部署 Microsoft Visual C++ 2005。这些必备项将启动服务器重新启动，并且 VDA 安装将在重新启动后继续运行。

有关详细信息，请参阅[使用独立的软件包安装 VDA](#)。

Delivery Controller 7.6.3 (Controller Hotfixes Update 3)

LTSR 版本：Delivery Controller 7.6.3

本版本最初是在 2015 年 11 月 12 日作为 Delivery Controller 7.6.3 (Controller Hotfixes Update 3) 的一部分发布的

系统要求

已修复的问题 (32 位)

已修复的问题 (64 位)

安装/升级

如果您的 Controller 的版本为 7.6.3：

Controller 7.6.3 (Controller Hotfixes Update 3) 为 LTSR 版本。如果您之前已升级到版本 7.6.3，则 Controller 符合 LTSR，不需要进行升级，可以跳至 Citrix Studio 部分。重要：请务必安装版本 7.6.3 的所有组件；否则，您的 Controller 可能会保留在不稳定状态。

如果您的 Controller 的版本为 7.6、7.6.1 或 7.6.2：

要符合 LTSR，需要将您的 Controller 升级到 LTSR 版本。为此，请将 LTSR 版本下载到您的 Controller 并按照下面的升级说明进行操作。

警告：不支持从各个 Controller 组件降级（又称为回滚），并且降级可能会将您的系统保留在不稳定状态。Controller 组件不修补现有安装，每个组件都会使用新安装完全替换原始组件。因此，卸载某个组件会从 Controller 中删除整个组件。如果需要还原到早期版本的 Controller，则必须卸载每个组件，然后重新安装每个组件的早期版本。还原到早期版本的组件可能会导致丢失您在安装此升级时配置的设置。

重要：必须安装 LTSR 版本的所有组件；否则，您的 Controller 可能会保留在不稳定状态。

如果要从 7.6 Controller 的基础版本 (RTM) 进行升级，请安装 LTSR Controller 的所有组件。

如果要从 Delivery Controller 7.6.1 (Controller Hotfixes Update 1) 或 Delivery Controller 7.6.2 (Controller Hotfixes Update 2) 进行升级，请仅安装 LTSR 版本中的新组件（与已安装的早期版本相比）。不需要按照特定顺序安装各个组件。

- 要成功升级，服务器不得设置注册表修改限制。
- 有关安装 XenDesktop/XenApp 7.x Controller 更新的补充信息，请参阅 [CTX201988](#)。

警告。根据设计，Broker Service (BrokerSvc760WX64003.msi) 组件会修改您的站点数据存储的 Broker DbSchema。这些修改具有永久性和不可逆性。如果您因任何原因决定稍后卸载 Broker Service 组件，这些修改不会自动还原。以防万一，Citrix 强烈建议您先备份站点数据存储，然后再安装 Broker Service 组件。这样将允许您将站点数据存储手动还原到备份的版本。即使这样，您在备份和还原站点数据存储期间对其所做的所有更改都将丢失。有关备份和还原数据存储的信息，请参阅 [CTX135207](#)。

仅当至少创建了一个站点时，才能成功更新 DbSchema。如果尚未创建站点，请先至少创建一个站点，然后再安装此更新。否则，安装将无法更新现有 DbSchema，并且您需要重新构建 XenDesktop。

注意：升级到本版本后，系统将显示一条针对 Citrix Studio 中的许可证服务器兼容性检查的提示，用于确保您的许可证服务器为所需版本。如果使用的是随 XenDesktop 7.6 发布的许可证服务器或更新版本中的许可证服务器，则不需要升级许可证服务器。单击“继续”以继续升级 DBSchema。

1. 务必确认从 7.6.2 Controller 进行升级。否则，请参阅下文“从 Delivery Controller 7.6.1 或 7.6 进行升级”。
2. 确保至少存在一个站点。
3. 备份您的站点数据存储。
4. 将发行包复制到网络上的共享文件夹。
5. 保存要更新的 Delivery Controller 上的组件 msi 文件。
6. 运行 .msi 文件。
7. 重新启动 Delivery Controller（即使系统未提示也需要重新启动）。
8. 要升级到本版本安装的最新 DbSchema，请转至 Citrix Studio 的“控制板”，然后单击“升级”。

注意。本部分不适用于从 Controller 7.6.2 进行升级。

警告。根据设计，Broker Service (BrokerSvc760WX64003.msi) 和 Host Service (HostSvc760WX64003.msi) 组件会分别修改您的站点数据存储的 Broker 和主机 DbSchema。这些修改具有永久性和不可逆性。如果您因任何原因决定稍后卸载 Broker Service 或 Host Service 组件，这些修改不会自动还原。以防万一，Citrix 强烈建议您先备份站点数据存储，然后再安装 Broker Service 和 Host Service 组件。这样将允许您将站点数据存储手动还原到备份的版本。即使这样，您在备份和还原站点数据存储期间对其所做的所有更改都将丢失。有关备份和还原您的数据存储的信息，请参阅 [CTX135207](#)。

仅当至少创建了一个站点时，才能成功更新 DbSchema。如果尚未创建站点，请先至少创建一个站点，然后再安装此更新。否则，安装将无法更新现有 DbSchema，并且您需要重新构建 XenDesktop。

注意。升级到本版本后，系统将显示一条针对 Citrix Studio 中的许可证服务器兼容性检查的提示，用于确保您的许可证服务器为所需版本。如果使用的是随 XenDesktop 7.6 发布的许可证服务器或更新版本中的许可证服务器，则不需要升级许可证服务器。单击“继续”以继续升级 DBSchema。

1. 确保至少存在一个站点。
2. 备份您的站点数据存储。
3. 将发行包复制到网络上的共享文件夹。
4. 保存要更新的 Delivery Controller 上的组件 msi 文件。

5. 运行 .msi 文件。
6. 重新启动 Delivery Controller（即使系统未提示也需要重新启动）。
7. 要升级到本版本安装的最新 DbSchema，请转至 Citrix Studio 的“控制板”，然后单击“升级”。

1. 从“ARP”/“程序和功能”中卸载组件。
2. 按 [CTX135207](#) 中所述还原数据存储。
3. 安装所需的组件版本（基础版本或更高版本）。
4. 重新启动 Controller（即使系统未提示也需要重新启动）。

Citrix Studio 7.6 Update 3

LTSR 版本：Citrix Studio 7.6 Update 3

本版本最初是在 2015 年 10 月 29 日作为修补程序 DStudio760WX64003 发布的；修补程序 DStudio760WX86003

系统要求

已修复的问题

- [64 位](#)
- [32 位](#)

已知问题

如果升级过程中 Citrix Studio 处于打开状态，并且您在此修补程序的安装向导的“正在使用的文件”页面上选择设置“关闭应用程序并尝试重新启动”，则可能会显示以下错误消息：

Setup was unable to automatically close all requested applications. Please ensure that the applications holding files in use are closed before continuing with the installation.（安装程序无法自动关闭所有请求的应用程序。请务必先关闭正在使用文件的应用程序，然后再继续安装。）

如果显示此消息，您可以安全地将其关闭并单击“确定”以继续安装。

安装/升级

下载 Citrix Studio 的 LTSR 版本并按照 [CTX201572](#) 中提供的安装说明进行操作。

Citrix Director 7.6.300

LTSR 版本：Director 7.6.300

本版本最初是在 9 月 30 日作为 Feature Pack 3 的一部分发布的 (Director_7.6.300.zip)

系统要求

请确保您已在 IIS 中选中所有必需的功能。有关完整列表，请参阅 [CTX142260](#)。安装 Citrix Group Policy Management 组件（如果尚未安装）。

已修复的问题

安装/升级

将 Citrix Director 的 LTSR 版本下载到运行 Director 的服务器并按照 [Director](#) 上的说明进行操作。

Group Policy Management 7.6.300 (2.5)

LTSR 版本：Group Policy Management 7.6.300

本版本最初是在 9 月 30 日作为 Feature Pack 3 的一部分发布的 (CitrixGroupPolicyManagement_7.6.300.zip)

系统要求：

运行 Windows 7、Windows 8、Windows 8.1、Server 2008 R2、Server 2012 或 Server 2012 R2 的计算机

VDA 中新增和增强的 HDX 技术功能使用更新后的 Group Policy Management 包进行管理。注意：安装后，此组件在“程序和功能”中显示为版本 2.5.0.0。

安装/升级

需要在安装了 Director 的系统中安装 Citrix Group Policy Management，以使策略在“用户详细信息”视图中显示。在运行 Director 的服务器上下载并安装 Citrix Group Policy Management (Citrix Policy) 的 LTSR 版本。下一步，请启动 Studio 或 GPMC，此时将显示新的以及更新后的策略。

有关更新后的策略的详细信息，请参阅[视觉显示策略设置](#)，了解增强的 Thinwire 兼容模式；参阅[USB 设备策略设置](#)，了解签名设备和绘图平板电脑支持；参阅[Flash 重定向](#)和[多媒体策略设置](#)，了解视频回退防护。

StoreFront 3.0.1

LTSR 版本：3.0.1

本版本最初是在 2015 年 9 月 30 日作为 Feature Pack 3 的一部分发布的 (CitrixStoreFront-x64.exe)

[系统要求](#)

[已修复的问题](#)

[已知问题](#)

安装/升级

将 StoreFront 的 LTSR 版本下载到 StoreFront 服务器并按照[升级说明](#)进行操作。

Provisioning Services 7.6 Update 1

LTSR 版本：Provisioning Services 7.6 Update 1 (面向服务器和控制台的 Provisioning Services 7.6 Cumulative Update 1) ； PVS760TargetDeviceWX64001.zip、PVS760TargetDeviceWX86001.zip

本版本最初是在 2015 年 9 月 15 日作为 PVS760ConsoleServerWX86001.zip 发布的；PVS760ConsoleServerWX64001.zip

Provisioning Services 7.6 Update 1 中包含在基础 7.6 版本中发现的 40 多个问题的修复。

[系统要求](#)

[已修复的问题](#)

- 控制台、服务器 ([64 位](#) | [32 位](#))
- 目标设备 ([64 位](#) | [32 位](#))

安装/升级

下载 Provisioning Services 的 LTSR 版本并按照[安装 Provisioning Services 控制台软件](#)（控制台）、[安装 Provisioning Services 服务器软件](#)（服务器）和 [CTX135746](#)（目标设备）中的安装说明进行操作。

Session Recording 7.6.100

LTSR 版本：7.6.100

本版本最初是在 2015 年 6 月 30 日作为 Feature Pack 2 的一部分发布的 (SessionRecording7.6.100.zip)

作为 Feature Pack 2 的一部分发布的 Session Recording 7.6.100 包括以下新增功能和增强功能。

- 可以在安装 Session Recording 数据库组件时指定数据库的连接凭据。
- 可以在安装 Session Recording 数据库和 Session Recording Server 组件时测试数据库的连接性，在安装 Session Recording Agent 组件时测试 Session Recording Server 的连接性。
- 安装 Session Recording 数据库无需具备 Microsoft Shared Management Objects。
- Citrix 体验改善计划 (CEIP) 集成到 Session Recording 中。有关详细信息，请参阅关于 Citrix 客户体验改善计划。升级过程中保留现有设置。

系统要求

安装/升级

下载 LTSR 版本，然后按照[升级说明](#)进行操作。

在 LTSR 部署中使用 Session Recording 7.6.100 的已知问题：

- Microsoft 画图会话的录制件无法在 Session Recording Player 中正确播放。[#0604700]
- 播放在多监视器用户设备上录制的会话时出现错误。[#0605129]

通用打印服务器 7.6.300

LTSR 版本：7.6.300

本版本最初是在 2015 年 9 月 30 日作为 Feature Pack 3 的一部分发布的 (UpsServer_7.6.300.zip)

注意：通用打印服务器由客户端和服务端组件组成。客户端组件作为 VDA 的一部分进行安装；因此，LTSR 版本中不包含任何客户端安装文件。在服务器端，LTSR 不支持在 32 位 Windows 操作系统中安装通用打印服务器；因此，仅包含 64 位服务器安装程序。

系统要求

已修复的问题

安装/升级

通用打印服务器包中包含更新版本的独立 UPS 服务器组件 (UpsServer_x64.msi) 和必备的 vcredist_x64.exe、vcredist_x86.exe 和 cdf_x64.msi 文件。

1. 将 LTSR 版本下载到 Windows 2008 R2 SP1、Windows Server 2012 或 Windows Server 2012 R2 打印服务器。
2. 安装必备的 vcredist_x64.exe、vcredist_x86.exe 和 cdf_x64.msi 文件。

3. 安装通用打印服务器组件 UpsServer_x64.msi。
4. 安装通用打印服务器组件后重新启动服务器。

UPClient 组件属于 VDA 安装的一部分。因此，不需要手动安装客户端组件，并且客户端组件不作为 LTSR 版本的独立组件包括在内。

有关详细信息，请参阅[置备打印机](#)。

HDX Flash 重定向

HDX Flash 重定向功能可将大部分 Adobe Flash 内容（包括动画、视频和应用程序）处理工作转移到连接至 LAN 和 WAN 的用户 Windows 设备，从而降低服务器和网络的工作负载。这样将提供更高的可扩展性，同时确保获得高清晰度用户体验。

客户端组件作为桌面和服务器操作系统 VDA 的一部分进行安装。因此，升级到 VDA 的 LTSR 版本会将您的部署升级到最新版本的 HDX Flash 重定向。

不需要安装任何服务器端组件。但是，配置 Flash 重定向时，必须同时设置服务器端和客户端。有关配置 Flash 重定向的信息，请参阅[Flash 重定向](#)。有关 HDX Flash 最新更新的兼容性，请参阅[CTX136588](#)。

Feature Pack 3：安装和升级

May 04, 2016

本文提供 XenApp 和 XenDesktop 7.6 FP3 中的所有组件的简要安装说明，即，要安装的组件和安装位置。有关详细说明，请查看[系统要求](#)和产品文章链接。

从 citrix.com 下载 FP3 组件。可用性取决于您当前的授权。

在要安装 VDA 的计算机上下载并运行 **VDAWorkstationSetup.exe**。请使用图形界面或命令行。

有关详细信息，请参阅[使用独立的软件包安装 VDA](#)。

要将运行 Windows 8.x 或 Window 7 的计算机上安装的 VDA 升级到 Windows 10，请先对 Windows 7 和 Windows 8.x 计算机重新映像到 Windows 10，然后使用独立的软件包安装支持的适用于 Windows 10 的 VDA。

有关详细信息，请参阅[升级部署](#)。

在要安装 VDA 的计算机上下载并运行 **VDA ServerSetup.exe**。请使用图形界面或命令行。

VDA for Windows Server OS 安装会自动部署 Microsoft Visual C++ 2013 Runtime（32 位和 64 位）以及 2008 和 2010 Runtime（32 位和 64 位）。不再部署 Microsoft Visual C++ 2005。这些必备项将启动服务器重新启动，并且 VDA 安装将在重新启动后继续运行。

有关详细信息，请参阅[使用独立的软件包安装 VDA](#)。

本版本中新的和增强的 HDX 技术功能包含在 Feature Pack 3 VDA 中，并使用更新后的 Group Policy Management 软件包进行管理。

在管理 Citrix 策略的服务器上，下载并运行 **CitrixGroupPolicyManagement_x64.msi** 或 **CitrixGroupPolicyManagement_x86.msi**。下一步，请启动 Studio 或 GPMC，此时将显示新的以及更新后的策略。

有关更新后的策略的详细信息，请参阅[视觉显示策略设置](#)，了解增强的 Thinwire 兼容性模式；参阅 [USB 设备策略设置](#)，了解支持签名设备和绘图平板电脑；参阅 [Flash 重定向](#)和[多媒体策略设置](#)，了解视频回退防护；参阅 [Framehawk 策略设置](#)，了解 HDX Framehawk 虚拟通道。

包含用于为 Active Directory 中的几组计算机安装、升级或删除 Virtual Delivery Agents (VDA) 的脚本（**InstallVDA.bat** 和 **UninstallVDA.bat**）。使用之前必须修改这些脚本。

有关详细信息，请参阅[使用脚本安装或删除 Virtual Delivery Agent](#)。

UPS 软件包包含更新后的独立 UPS 服务器组件版本（**UpsServer_x64.msi** 和 **UpsServer_x86.msi**）以及必备项（**vcredist_x64.exe**、**vcredist_x86.exe**、**cdf_x64.msi** 和 **cdf_x86.msi**）。

在 Windows Server 2012 R2 或 Windows Server 2012 打印服务器上，通过提取并启动组件的 MSI、UpsServer_x64.msi 或 UpsServer_x86.msi 安装 UPServer 组件。安装 UPServer 组件后需要重新启动。

安装在 XenApp 和 XenDesktop 主机（负责置备会话网络打印机）上的 UPClient 组件属于 VDA 安装的一部分。

对于希望单独部署 UPClient 组件的环境（例如部署了 XenApp 6.5 的环境），需要执行额外的步骤。

有关 FP3 中的所有配置的详细信息，请参阅[置备打印机](#)。

在 Windows 2008 32 位平台上部署 UpsServer_x86.msi 也需要执行额外的步骤。有关详细信息，请参阅[使用命令行安装](#)。

请确保您已在 IIS 中选中所有必需的功能。有关完整列表，请参阅 [CTX142260](#)。在运行 Director 的服务器上下载并运行 **DesktopDirector.msi**。安装 CitrixGroupPolicyManagement.msi（如果尚未安装）。

使用此软件包中包含的 WMIProxy MSI 文件安装或升级 VDA 上的 WMIProxy。

使用此软件包中包含的 XDPoshSnapin MSI 在 Delivery Controller 上的 XDPoshSnapin_Hotfix。委派管理员和自定义管理员要想查看 Director 中的 Framehawk 虚拟通道信息，需要执行此操作。

有关完整的安装和配置说明，请参阅 [Director](#)。

AppDNA 7.6.5 包括新的 Windows 10 报告，以帮助您了解 Windows XP 或更高版本中的应用程序是否将在 Windows 10 上运行。下载安装程序 **Citrix-AppDNA.msi** 并在 VM 中运行。该安装程序首次处理并升级所有许可类型的安装。对于试用版许可证，安装程序将创建一个能够在 Microsoft SQL Server Express 中运行的数据库。

有关详细信息，请参阅[安装 AppDNA](#)。

要进行安装，请以管理员身份在 StoreFront 服务器上下载并运行 **CitrixStoreFront-x64.exe**。StoreFront 3.0.1 包括 Citrix Receiver for HTML5 1.8。

有关详细信息，请参阅[安装、设置和卸载](#)。

要进行升级，请禁用对 StoreFront 部署的访问权限并重新启动 StoreFront 服务器。以管理员身份在 StoreFront 服务器上下载并运行 **CitrixStoreFront-x64.exe**。

有关详细信息，请参阅[升级](#)。

Receiver for HTML5 托管在 StoreFront 上，默认情况下，StoreFront 中包含某个版本的 Receiver for HTML5。StoreFront 3.0.1 包括 Citrix Receiver for HTML5 1.8。如果您运行的是早期版本的 StoreFront，希望安装或升级到此版本的 Citrix Receiver for HTML5，可以在 StoreFront 2.5 或更高版本上下载并运行 Receiver for HTML5 1.8 (**CitrixHTML5Client-x64.exe**)。

有关详细信息，请参阅[配置 Citrix Receiver for HTML5](#)。

Receiver for Windows 4.3.100 包含在随 FP3 交付的 VDA 7.6.300 软件包中。请使用 Citrix Receiver for Windows 4.3.100 (CitrixReceiver.exe) 在早期版本的 VDA 上升级 Receiver，或者在用户设备上安装或升级。

如果 Citrix Lync Optimization Pack 安装在端点设备上，则必须在升级 Citrix Receiver for Windows 后卸载并重新安装该软件。请参阅 [CTX200340](#)。

有关详细信息，请参阅[安装](#)。

准备您的 Linux 虚拟机，在其上安装新的 [Linux VDA 软件](#)，配置 Delivery Controller，然后使用 Studio 向用户提供这些桌面。

有关详细信息，请参阅 [Linux Virtual Desktop Installation Guide for SUSE](#)（《面向 SUSE 的 Linux 虚拟桌面安装指南》）以及 [Linux Virtual Desktop Installation Guide for Red Hat](#)（《面向 Red Hat 的 Linux 虚拟桌面安装指南》）。

这是面向运行 XenServer 6.5.0 Service Pack 1 的客户的修补程序 此修补程序提供面向标准 Windows 驱动程序的 Windows 10 认证。

有关详细信息，请参阅 [CTX201974](#)。

Feature Pack 3 : 已修复的问题

Jul 13, 2016

- Director 中的 Activity Manager 可能无法显示为某些用户打开的应用程序。
[#LC2325]
- 配置了多个站点时，Director 站点名称被截断。
[#LC2967]

通用打印客户端

通用打印服务器

通用打印客户端问题

- 使用网络打印机时，打印预览显示 Office 文档的速度可能会非常缓慢。
[#LC1205]
- 在安装了 XenApp 6.5 Hotfix Rollup Pack 4 和通用打印客户端 7.6 的服务器上，用户设备上的会话可能会无响应。
[#LC1562]
- 此功能增强了通用打印服务器和通用打印客户端的实现性能计数器。
[#LC1820]
- 尝试从 Adobe Reader X1 打印时，打印对话框可能需要长达 20 秒的时间才能显示。
[#LC2412]
- 用户登录到 VDA 时，Citrix Print Manager Service (cpsvc.exe) 有时可能会意外停止运行。
[#LC2716]
- VDA 上安装了通用打印服务器但被 Citrix 策略禁用时，VDA 上的 PowerShell get-printer 命令可能会失败。
[#LC2966]
- 打印后台处理程序服务有时会在 UpProv.dll 上遇到致命异常。
[#LC3055]

通用打印服务器问题

- 此修复解决了基础组件中的内存问题。
[#LC1381]

- 此功能增强了通用打印服务器和通用打印客户端的实现性能计数器。

[#LC1820]

- 此修复解决了基础组件中的内存问题。

[#LC2764]

XenApp 和 XenDesktop 7.6 Feature Pack 3 (7.6.300) 与 7.6 LTSR 中包含的 VDA for Desktop OS 的版本相同。XenApp 和 XenDesktop 7.7 中包含的版本包含的修复与版本 7.6.300 相同，并且还包含 XenApp 和 XenDesktop 7.7 的兼容性更新。

Broker Agent	Remote PC
HDX 3d Pro	无缝窗口
HDX MediaStream Flash 重定向	会话/连接
HDX MediaStream Windows Media 重定向	站点/场管理
安装、卸载、升级	智能卡
键盘/鼠标	系统异常
本地应用程序访问	用户体验
登录/身份验证	用户界面
打印	其他

Broker Agent

- VDA 可能会卡在注册过程的初始化状态。Citrix Desktop Service 在未经过重新启动的情况下运行数日后会发生此问题。
[#LC0570]
- 启用函数“CName”时，VDA 注册可能需要非常长的时间。应用此修复后，打开 CNameSuppressOrgFqdnLookup 值会加快注册过程的速度。
重要：如果启用了 CNameSuppressOrgFqdnLookup，则必须启用 CName。

要启用此修复，必须创建以下注册表项：

- HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\VirtualDesktopAgent
名称：UseCnameLookup
类型：REG_DWORD
值：已启用 = 1 (已禁用 = 0)

- HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\VirtualDesktopAgent

名称：CNameSuppressOrgFqdnLookup

类型：REG_DWORD

值：已启用 = 1 (已禁用 = 0)

[#LC0791]

- 如果启用了 VDI-in-a-Box 模式，尝试通过用户会话重新连接到 VDA 可能会失败。

[#LC1522]

- 如果 VDA 版本 7.6 是 XenDesktop 5.6 站点的成员，当用户在连接后又断开连接，然后再重新连接时，重新连接尝试将失败，且 VDA 未注册。

[#LC1859]

HDX 3D Pro

- 用户在 HDX 3D Pro 桌面中加载自定义光标时，不显示正确的光标颜色。

[#LC0917]

- 在第三方 CAD 应用程序中，如果桌面启用了 HDX 3D Pro 且运行的是简体中文版的 Windows，当在其工具栏和绘图区域上方移动鼠标光标时，鼠标光标可能会消失。

[#LC1321]

- Citrix Receiver 会话断开连接后，在使用 HDX 3D Pro 和 NVIDIA 图形卡的计算机上，重新连接到锁定的 VDA 会话可能会失败。

[#LC1629]

- 在校准到 150% DPI 的双显示器 HDX 3D Pro VDA 会话中，鼠标指针和其他屏幕艺术效果可能会显示到 VDA 会话外部，因此无法访问。

[#LC2502]

- 启用硬件加速时，重新调整 CD Viewer 窗口的大小后 Receiver 会话中的 AutoCAD 2011 失败。

[#LC2531]

- 应用此修复后，HDX 网络通信的带宽容量为 20 兆位/秒 (Mbps)。

[#LC2780]

- 在使用 NVidia Grid K1 图形卡的 HDX 3D Pro 环境中，如果 VDA 连接到高延迟 WAN 网络，VDA 会话中的图形可能会模糊。

[#LC2877]

- 在启用 HDX 3D Pro 的 VDA 中，无损指示器任务栏图标（适用于 LLIndicator.exe）可能无法指示无损状态，并且可能会在单击鼠标右键时无响应。

[#LC2931]

- 用户在 HDX 3D Pro 桌面上多次锁定或解锁会话后，鼠标光标可能会消失。

[#LC3188]

- Desktop Viewer 从全屏模式切换到窗口模式时，HDX 3D Pro 会话可能会在几分钟之内无响应。

[#LC3193]

- HDX 3D Pro 可能会在 VDA 上意外启动，尽管并未安装此程序。在 VDA 上安装 Windows XP 显示驱动程序模型 (XPDM) 而非 Windows 显示驱动程序模型 (WDDM) 时会发生此问题。

[#LC3215]

HDX MediaStream Flash 重定向

- 使用 Microsoft Internet Explorer 9 或此浏览器的更高版本时，如果在另一个 Flash 重定向视频正在播放期间单击某个 Flash 重定向视频的链接，两个视频的视频流会重叠。

此修复并未消除此问题，但是通过在服务器上设置以下注册表项，可以缩短此问题持续的时间：

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer

名称：UrlListForTerminationFlashInst

类型：REG_MULTI_SZ

数据：<受影响的 Web 站点的 URL；同时包括 http:// 和 https://，每个 URL 单独占用一行；例如：

http://www.youtube.com/; https://www.youtube.com/>

[#LC0453]

- 使用 Microsoft Internet Explorer 9 或更高版本时，Web 站点中嵌入的视频的 Flash 重定向可能会失败。视频所在位置会显示黑色长方形。

要启用此项修复，必须设置以下注册表项：

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer

名称：WorkWithAeroEnabled

类型：REG_DWORD

数据：0

[#LC0477]

- 在 Microsoft Internet Explorer 8 或此浏览器的更高版本中使用 YouTube 的内置“Search”（搜索）字段时，HDX Flash 重定向恢复为服务器端呈现。

要启用此项修复，必须在服务器上设置以下注册表项：

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer

名称：SupportedUrlHeads

类型：REG_MULTI_SZ

数据：<每个值单独占用一行，不分隔>

http://

https://

file://

注意：SupportedUrlHeads 最初是在修复 #LA4151 中引入的。此修复也使用该注册表项。如果向注册表中添加 SupportedUrlHeads，Flash 重定向将忽略 Flash 实例化请求，除非目标 URL 的开头为 SupportedUrlHeads 中指定的其中一个标头。

[#LC0505]

- cnn.com 和 msn.com 上嵌入视频的 Flash 重定向失败。视频所在位置会显示黑色长方形。

为启用此项修复，必须设置以下注册表项：

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer\ASJSCallbacks

对于 cnn.com：

名称：cnn.com

类型：REG_MULTI_SZ

数据：	javaScriptLoaded
	setQueueAutoplay
	setAdVisibility
	setPlayerMode
	setAdSection
	setAdKeyValue
	playContent
	getContentEntry
	switchTrackingContext

对于 msn.com：

名称：msn.com

类型：REG_MULTI_SZ

数据：	MsnVideoCallback
	MsnVideoPropertyCallback
	GetWidgetID

注意：此修复仅适用于 Internet Explorer 9 或更高版本。

[#LC0665]

- 使用 Microsoft Internet Explorer 10 时，cnn.com Web 站点中嵌入视频的 Flash 重定向可能会失败。如果此 Web 站点上的第一个视频自动播放，将会出现此问题。此视频的页面上的一些内容（如标题、状态和广告）可能会缺失。

[#LC0830]

- m.mlb.com 和 msn.com 上嵌入视频的 Flash 重定向失败。视频所在位置会显示黑色长方形。

要启用此修复，请设置以下注册表项：

对于 mlb.com：

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer\ASJSCallbacks

名称：m.mlb.com

类型：REG_MULTI_SZ

数据：<将每个值单独放在一行；不分隔>

setPlaylist

startPlaylist

对于 msn.com：

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer\ASJSCallbacks

名称：msn.com

类型：REG_MULTI_SZ

数据：<将每个值单独放在一行；不分隔>

MsnVideoCallback

MsnVideoPropertyCallback

GetWidgetID

此修复仅适用于 Internet Explorer 9 或更高版本。要使此设置生效，请将 Internet Explorer 设置为以模仿模式运行，将 Internet Explorer 9 和 10 版设置为以文档模式运行。有关详细信息，请参阅 Citrix 支持站点上的文章 [CTX136588](#)。

[#LC1043]

- 在下列情况下，尝试播放 Web 站点上的视频时，Internet Explorer 可能会异常关闭：
 - 启用选项“禁用 Flash 加速”，或者站点位于黑名单上。
 - 用户使用的是不支持 Flash 重定向的 Receiver 版本，如 Receivers for Mac OS X、iPad、iPhone 和 Android。

[#LC2256]

HDX MediaStream Windows Media 重定向

- 此版本修复了 HDX 图形编码器生成的冗余帧，实现了更为顺畅的帧速率和更好的交互性。

[#LC2177]

- 在使用 Receiver for Windows 13.3 或 13.4 版本的 VDA 会话中播放视频文件时，自定义应用程序中出现音频和视频同步。但是，升级到更高版本的 Receiver 后，Windows Media 重定向失败。出现此问题后，用户只能听到声音或只能看到视频。

要启用此修复，必须使用 Citrix Receiver for Windows 4.3。

[#LC2516]

- 尝试关闭其中的视频已暂停的 Windows Media Player 会导致 Windows Media Player 无响应。

[#LC0728、#LC3410]

安装、卸载、升级

- VDA for Desktop OS 的早期更新尝试安装 WDDM 驱动程序。如果在 VDA 初始安装期间曾使用 /nocitrixwddm 开关阻止安装 WDDM 驱动程序，安装更新会导致安装过程循环进行，从而使 VDA 不断重新启动。应用此修复后，在最初阻止安装此驱动程序的情况下，更新 VDA 时不会再尝试安装此驱动程序。

[#LC0443]

- 安装任何在 ICAWS750WX64012 之前发布的 VDA 修补程序会将 Citrix 远程多点触控设备和 Citrix Multi-Touch Redirection Services 的状态还原为默认值，不管您之前是否手动启用或禁用了这些值。

[#LC0446]

- 安装 2014 年 9 月之前发布的适用于 Windows 桌面操作系统和服务器操作系统的 XenApp 7.5、XenDesktop 7.1 和 7.5 VDA Core Services 的修补程序，会导致 ICA 会话性能监视器计数器被删除。这会对倚赖这些计数器的工具和进程的运行产生不利影响。此修复恢复了因为安装早期修补程序而被删除的计数器，并阻止在安装后续修补程序时将其删除。

[#LC0771]

- 将 VDA 从 7.6 版降级到 5.6.400 版后，VDA 可能无法注册。

[#LC1465]

- 升级 VDA 之后，出现以下问题：
 - 注册表项 EnableReadImageFileExecOptionsExclusionList 更改为“outlook.exe”。
 - Outlook 无法使用智能卡。此修复从注册表 EnableReadImageFileExecOptionsExclusionList 中删除了“outlook.exe”。此注册表项中的所有其他字符串保留。

[#LC1655]

- 用户将 VDA 升级到版本 5.6.400 或 5.6.500 后，将文件从映射的客户端驱动器复制到 VDA 或直接从驱动器打开这些文件时，显示以下错误消息：

“The Handle is Invalid.”（句柄无效。）

[#LC1910]

- 尝试通过使用 Services 管理单元重新启动 Print Manager Service 失败。显示错误消息“Couldn't stop this service”（无法停止此服务），且 Print Manager Service 的状态显示为“stopping”（正在停止）。

[#LC2122]

- 将 VDA 从 XenDesktop 7.5 升级到 XenDesktop 7.6 后，用户尝试进行连接时，显示错误 1030。系统事件日志中显示以下消息：“The Citrix ICA Transport Driver received an invalid Transport packet on port 2598.”（Citrix ICA 传输驱动程序在端口 2598 上收到无效的传输数据包。）

[#LC2501]

- VDA for Desktop OS 的早期更新尝试安装 WDDM 驱动程序。如果在 VDA 初始安装期间曾使用 /nocitrixwddm 开关阻止安装 WDDM 驱动程序，安装更新会导致安装过程循环进行，从而使 VDA 不断重新启动。应用此修复后，在最初阻止安装此驱动程序的情况下，更新 VDA 时不会再尝试安装此驱动程序。

[#LC3052]

- 在少数情况下，安装或重新安装 MSP 修补程序软件包会导致将 Citrix Display Driver (Citrix Systems – WDDM) 标记为删除。这会导致安装失败，从而使 VDA 的显示器分辨率不可用。

[#LC3312]

键盘/鼠标

- 在 iPad 设备上使用 OpenOffice 应用程序时，自动显示键盘功能不起作用。

[#LC2717]

本地应用程序访问

- 在全屏模式的桌面会话中，如果启用本地应用程序访问且禁用“拖动时显示窗口内容”设置，在 Internet Explorer 中的图片上方移动应用程序图标时，高分辨率图片会变形。

[#LC1722]

登录/身份验证

- 用户尝试通过使用智能卡身份验证的 StoreFront 启动应用程序时，显示错误消息“未找到有效证书”。这会阻止用户登录。

[#LC1704]

打印

- Citrix Print Manager Service (CpSvc.exe) 进程可能会意外退出。

[#LA5682、#LC1770]

- 设置时未添加网络打印机的“会话打印机”策略可能会阻止创建其他“会话打印机”策略中指定的会话打印机。

[#LC0705]

- 更改 VDA 中的默认打印机后，重新启动 VDA 时，如果用户未注销 ICA 会话，之前的默认打印机仍显示在 VDA 中。

[#LC0898]

- 此修复恢复了“旧版打印机名称”策略，此策略之前在该产品的 7.x 版本中不可用。

[#LC1002]

- 如果“pServerName”的值不正确，Citrix Print Manager Service (CpSvc.exe) 进程可能会意外退出。

[#LC1241]

- 启用了“自动创建一般通用打印机”策略设置时，Citrix Print Manager Service (CpSvc.exe) 将不创建 Citrix 通用打印机一般打印机。

[#LC1255]

- 此功能增强了通用打印服务器和通用打印客户端的实现性能计数器。

[#LC1820]

- 使用 Citrix 通用打印机驱动程序时，打印的第一页可能为空白。

[#LC2771]

- 即使将“通用打印 EMF 处理模式”策略设置成“为打印机重新处理 EMF”，此策略仍保留默认值设置。

[#LC2521]

- 用户尝试打印文档或选择其他打印机时，可能会遇到间断性的延迟，并在长达一分钟的时间内看到消息“Connecting to printer”（正在连接到打印机）。

[#LC2701]

- 应用修复 #LC0898 并配置会话漫游后，在客户端打印机和会话打印机中创建打印机失败。

[#LC3490]

Remote PC

- 如果在 RemotePC VDA 存在现有远程桌面和控制台会话的情况下，用户对同一 RemotePC VDA 启动 Receiver 会话，Receiver 会话将失败并显示消息“正在重新连接...”。此消息会一直存在，直到用户手动关闭 Receiver 会话为止。

[#LC0627]

- 如果您在重新启动后登录屏幕出现约 30 之内登录物理 Remote PC 控制台，将无法在原始会话持续期间与此 VDA 建立 Remote PC ICA 会话。

[#LC3603]

无缝窗口

- 移动某个停靠应用（如 Windows 任务栏）时，最大化的无缝应用程序不会正确调整大小。

[#LC1342]

- 尽管应用程序进程正在运行且服务器 VDA 上存在会话，无缝窗口仍不显示。

[#LC1728]

会话/连接

- 使用不同于初始会话的连接方法（例如，使用与不使用 Access/NetScaler Gateway）重新连接 VDA 会话时，不会为重新连接的会话重新评估 Citrix 策略。

注意：此修复要求您删除 Windows 桌面操作系统或 Windows 服务器操作系统设备上以下位置的所有条目。如果您没有受到所述问题的影响，请勿应用此修复：

- 注册表：HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Citrix
- 文件夹：C:\ProgramData\Citrix\GroupPolicy*.*

[#LA5051]

- 在具有相同屏幕分辨率的双显示器端点 A 和 B 场景中，从端点 A 断开连接，然后从端点 B 重新连接时，所有顶层窗口被重新放置到主显示器中。

[#LA5440]

- 打开已发布的应用程序，然后运行 explorer.exe 命令以打开 Windows 资源管理器会导致打开桌面而非 Windows 资源管理器。

注意：安装此修复后，explorer.exe 进程可能会意外退出。有关详细信息，请参阅知识中心文章 [CTX128009](#)。

[#LC0285]

- 重新连接到 Receiver 会话时，如果多个 Windows 资源管理器正在 VDA 上运行，CPU 使用率可能会很高。此外，最小化窗口不停地最大化，直到重新将其最小化到 Windows 任务栏才停止。

[#LC0315]

- 连接 USB 磁卡读卡器设备时，设备在虚拟桌面中可以被识别，但是未加载正确的驱动程序。

[#LC0360]

- 通过按 Windows+向左/向右箭头键将应用程序移动到屏幕边缘，然后再断开会话连接并重新连接会话时，应用程序的大小改变。大小改变的程度因应用程序而异，并且在主显示器上更明显。

[#LC0386]

- 有时，用户会话可能会变得无响应，并且无法注销。如果用户手动从控制台注销，也会发生此问题。

[#\$LC0472]

- 如果在 USB 重定向策略中禁用“简化设备连接”设置，SafeNet UKey USB 设备可能无法使用。

[#LC0480]

- 用户从任务栏关闭 Visual Basic Editor 的已发布版本时，打开的已发布 Excel 电子表格也关闭。

[#LC0534]

- 将 SD 卡插入用户设备，然后启动 VDA 会话时，如果通过客户端驱动器映射将此卡重定向，此卡在 Windows 资源管理器中显示为软盘。

要启用此修复，请设置以下注册表项：

- 在 32 位系统中：
HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\UNCLinks
名称：DisableFloppyIcon
类型：REG_DWORD
值：1
- 在 64 位系统中：
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\UNCLinks
名称：DisableFloppyIcon
类型：REG_DWORD
值：1

[#LC0616]

- 在用户设备上，尝试远程“区域和语言 > 日期和时间格式”下面的组合框控件可能会失败。如果组合框控件已经被成功远程过一次，第二次尝试远程此控件会发生此问题。

[#LC0836]

- 重定向的 COM 端口设备在 Receiver 会话中可能不可用。

[#LC0851]

- VDA 变得无响应时，从 Receiver 显示“Connection Established, Negotiating Capabilities”（已建立连接，正在协商功能）弹出窗口。要成功启动任何其他会话，需要重新启动 VDA。

[#LC0909]

- 用户使用 Receiver for Windows 登录时，时区不正确。要使此修补程序正常运行，必须安装以下修补程序：
 - 在用户设备和服务器上安装相同的 Microsoft 时区更新修补程序。例如，如果用户设备上安装了 Microsoft 修补程序 [KB2998527](#)，服务器上也应安装此修补程序。
 - 如果服务器操作系统为 Windows Server 2008 R2 Service Pack 1，在此服务器上安装 Microsoft 修补程序 [KB2870165](#)。
 - 在用户设备上安装修复 #LC1392。

[#LC1061]

- 如果 USB 重定向策略包含的字符数超过 1000，所有 USB 驱动器都会重定向，即使存在设备的拒绝规则也是如此。

[#LC1153]

- 如果在 Microsoft Project 2013 中启用 ExcelHook 并且打开多个项目，打开的第一个项目可能不会显示在任务栏中。

[#LC1158]

- 在广域网 (WAN) 连接上使用桌面组合重定向 (DCR) 时，显示文本和图形内容的 Web 页面或文档可能无响应，并且可能会占用大量带宽。

[#LC1256]

- 如果在 Desktop Delivery Controller (DDC) 上启用“设为无损”和“允许视觉无损压缩”设置，最终图像质量可能会降级。

[#LC1271]

- 用户首次登录时，在 Receiver 会话中屏幕可能会显示已损坏。将视觉质量更改为“设为无损”并启用“允许视觉无损压缩”设置时会出现此问题。

[#LC1272]

- 双显示器设置用于服务器 VDA 会话时，连接或取消连接辅助显示器可能会导致主显示器无响应。

[#LC1359]

- 用户尝试连接到已断开连接的会话时，建立连接需要长达三分钟时间。

[#LC1376]

- 用户无法从远程桌面 (RDP) 会话切换到 Receiver 会话。

[#LC1383]

- 在 XenDesktop 和 XenApp 会话中，重定向可能无法正确用于 MailTo 功能。用户单击 Web 浏览器中的“MailTo”链接时，不是打开已发布的 Microsoft Outlook，而是显示错误。

[#LC1400]

- 启动已发布的应用程序时，Citrix Graphics (CtxGfx.exe) 进程导致 CPU 使用率高于正常值。

[#LC1404]

- 启用会话延迟并在 iOS 或 Android 设备上关闭某个应用程序时，显示黑色窗口，且此窗口保留在设备屏幕上。

[#LC1407]

- 当用户设备上的 USB 存储设备连接到 XenDesktop 5.6 时，如果“UNCEnabled”的注册表值设置为“0”，“InitialClientDriveLetter”的注册表值设置为“N”，会话中 USB 磁盘的驱动器盘符开头为默认值“V”而非“N”。

有关详细信息，请参阅知识中心文章 [CTX127968](#) 和 [CTX122061](#)。

[#LC1451]

- 用户从桌面会话注销时，WFSHELLWindow 停止响应，且消息“Wfshell needs to close”（Wfshell 需要关闭）显示三分钟。

[#LC1591]

- VDA 进入屏幕保护或节能模式后，屏幕上的信息仍可见，直到用户提供将会话更新为黑色屏幕的输入（鼠标或键盘）。通过 DWORD 值 HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\Graphics\SetDisplayRequiredMode = 0 在会话中启用屏幕保护或节能选项时会出现此问题。

[#LC1650]

- 在 XenDesktop 会话中，第三方应用程序在访问映射的硬件保护装置时可能会变为无响应。

[#LC1657]

- 如果在注册表中启用 UxPersistence 项，在会话断开连接又重新连接后，尝试使用 Windows 键和向左或向右箭头键将应用程序移动到屏幕边缘失败。

[#LC1684]

- 为 VDI 将 Receiver 连接切换到远程桌面 (RDP) 会话时，如果用户在 RDP 连接的转移登录对话框中回答“否”，与 Receiver 进行重新连接会失败，并显示状态 1030。

[#LC1696]

- 用户从 XenDesktop 注销时，在 XenCenter 或 vSphere 控制台中，用户桌面会话显示几秒钟。

[#LC1782]

- 用户使用智能卡登录后又注销，然后再使用用户名和密码登录时，连接启动，紧接着又断开连接。此情况会一直持续，直到用户重新使用智能卡连接。

[#LC1831]

- 如果用户设备的键盘布局与 VDA 的键盘布局不同，从 Receiver for Windows 启动虚拟桌面会话期间，显示可能会在大约 60 到 70 秒的时间内无响应。

[#LC1939]

- 在 Windows 服务器和桌面 VDA 中，WMI Provider Service 的 CPU 使用率可能较高，从而导致 VDA 变得无响应。同样，Receiver 连接也会失败。

[#LC2041]

- 当用户使用 Citrix Receiver for Windows 从会话断开连接，然后在屏幕分辨率更小的设备上使用 Citrix Receiver for Linux 或 Citrix Receiver for Windows CE 重新连接到相同会话时，会话屏幕分辨率可能无法调整大小。如果会话启用了“旧图形模式”策略设置，可能会发生此问题。

[#LC2079]

- 如果在装有 Receiver for iOS 的 iPad 上启动 Lotus Notes，此应用程序可能会意外关闭。

[#LC2124]

- 在 XenDesktop 会话中，会显示一个表示服务要显示消息的消息框。当用户单击“View the message”（查看消息）时，XenDesktop 会话显示黑屏。当 XenDesktop VDA 已使用会话 0 时会出现此问题。

[#LC2160]

- 如果启用旧图形模式，不会自动禁用鼠标指针阴影。

[#LC2222]

- 用户从会话断开连接时，Citrix WinFrame Shell (WFSHELL.exe) 导致较高的 CPU 使用率。

[#LC2235]

- 如果用户从 VDA 桌面会话内登录 Web Interface 并尝试启动相同的桌面，会话可能会断开连接，并且 VDA 变得无响应。结果导致用户可能无法启动桌面，并且必须重新启动 VDA。

[#LC2267]

- 如果用户设备的默认 Unicode 输入语言与 VDA 的输入语言不同，加载会话桌面期间，会话可能变得无响应。

[#LC2287]

- 从客户端驱动器映射 (CDM) 打开已发布的应用程序文件，然后在输入文本后保存此文件时，修改的时间戳未更新。

[#LC2305]

- 用户从会话断开连接后，Wfshell.exe 继续占用大量 CPU 资源。

[#LC2391]

- 启用 USB 重定向的情况下，尝试使用 Wacom 平板电脑及其手写笔失败。

[#LC2409]

- 如果用户设备的键盘布局与 VDA 的键盘布局不同，加载会话桌面期间，会话可能变得无响应。

[#LC2466]

- 在启用 RDP 8 且未使用网络级别身份验证 (NLA) 的情况下，从 Receiver 断开连接后，与远程桌面 (RDP) 的连接尝试失败。

[#LC2480]

- 如果用户使用单点登录通过启用 ICA 代理的 NetScaler Gateway 连接到 StoreFront，然后注销再重新登录，VDA 策略将不起作用。用户断开连接再重新连接后，策略可正常运行。

[#LC2577]

- 启用 Excelhook 时，在打开的 Excel 工作簿之间切换可能会导致选定的工作簿丢失焦点。此外，用户无法选择任何已打开的 Excel 工作簿，焦点保留在选择的上一个文件上。

#LC2589]

- 更新密码注册表项之后，用户无法启动应用程序。要应用此修复，需要安装新版本的 SSL SDK。

[#LC2729]

- 用户重新连接到 VDA 时，如果启用了“User Experience Persistence”（用户体验持久性），应用程序窗口可能处于不正确的状态，如最大化而非已恢复。

[#LC2769]

- 销售点设备通过 COM 端口重定向到共享桌面时，共享桌面上运行的第三方软件可能无法识别此设备。

[#LC2775]

- 在 Windows Phone 8.1 上打开 Internet Explorer 的已发布版本时，如果将设备转为水平位置，半个屏幕会显示黑色窗口。此外，用户还无法最大化 Internet Explorer 窗口。如果用户将设备转为垂直状态，屏幕显示正常。

[#LC2786]

- Microsoft Office 应用程序在 VDA for Windows Desktop OS 上运行时可能会意外关闭。

[#LC2878]

- 结合使用外部显示器和便携式计算机时，外部显示器的边角可能会显示白色正方形。当外部显示器的屏幕分辨率与主显示器的分辨率不同，且位于主显示器的左侧，其顶部边缘在主显示器的顶部边缘上方时，尤其会出现此问题。

[#LC3186]

- Microsoft Office 可能会由于与 Citrix API 挂接的交互偶尔意外停止运行。

[#LC3280]

- 禁用 USB 重定向的情况下，从已发布的应用程序断开连接可能会导致事件查看器中显示以下虚假事件 261：

“The Citrix Device Redirector service could not complete an IO operation with the Redirector Bus.”（Citrix Device Redirector Service 无法使用重定向程序总线完成 IO 操作。）

[#LC3439]

站点/场管理

- 具有所有权限的自定义管理员在运行 PowerShell 命令“Remove-XAServer”时无法从场删除脱机服务器，并收到拒绝访问的错误消息。

- [#LC0752]

- 如果服务器只具有只读域控制器的访问权限，策略处理可能需要很长时间。

[#LC0828]

- 如果要使用 RES 软件来管理部署并且启用了“Instant LogOff”（即时注销）策略，用户注销会话时，可能无法为 XenDesktop 控制器上的 Citrix Broker Service 更新会话信息，从而导致控制器反映错误的会话状态。

[#LC1173]

- 此修复允许为用户设备创建永久性注册表设置。

[#LC1827]

- 如果延迟较高，Group Policy Engine (CitrixCseEngine.exe) 可能会在登录时导致延迟。

要启用此修复，请创建以下注册表项：

- HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\GroupPolicy
名称：GpoCacheEnabled
类型：REG_DWORD
值：1
- HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\GroupPolicy
名称：CacheGpoExpireInHours
类型：REG_DWORD
值：5 到 24

[#LC1987]

智能卡

- 升级 VDA 后，用户尝试使用智能卡登录时，Microsoft Outlook 不会提示输入个人标识号 (PIN)，而是会显示以下错误消息：

“请插入智能卡。”

[#LC0556]

- 连接到端点的读卡器中存在智能卡时，使用 Citrix Receiver for Linux 进行身份验证后，会话可能会无响应。

[#LC0982]

- ActivIdentity (accompks.exe) 进程可能会意外退出并在 scardhook64.dll 上出现异常。结果导致需要智能卡的 Web 站点无法正常运行，直到重新启动 VDA 后才可以。

[#LC1056]

- 利用此增强功能，从端点移除智能卡读卡器与从端点移除智能卡产生的行为相同。结果导致将 VDA 智能卡策略配置为“锁定工作站”时，删除读卡器会断开会话连接，从而锁定 VDA。

要启用此修复，请设置以下注册表项：

- HKEY_LOCAL_MACHINE\Software\Citrix\SmartCard
名称：ReaderUnpluggedDetectEnable
类型：REG_DWORD
值：非零值
- HKEY_LOCAL_MACHINE\Software\Citrix\SmartCard
名称：ReaderUnpluggedDetectPollingInterval
类型：REG_DWORD
值：5 到 60 秒（默认轮询间隔为 10 秒）

[#LC1059]

- 此修复扩展了修补程序 #LA1059 引入的功能，因此此修复对 Windows 8.1 端点到 VDA 7.5 版本的连接也有效。

根据 #LA1059 的说明：

利用此增强功能，从端点移除智能卡读卡器与从端点移除智能卡产生的行为相同。结果导致将 VDA 智能卡策略配置为“锁定工作站”时，删除读卡器会断开会话连接，从而锁定 VDA。

要启用此修复，请设置以下注册表项：

- HKEY_LOCAL_MACHINE\Software\Citrix\SmartCard
名称：ReaderUnpluggedDetectEnable
类型：REG_DWORD
值：非零值
- HKEY_LOCAL_MACHINE\Software\Citrix\SmartCard
名称：ReaderUnpluggedDetectPollingInterval
类型：REG_DWORD
值：5 到 60 秒（默认轮询间隔为 10 秒）

[#LC1713]

- VDA 服务器在 icaridd.dll 上遇到错误，显示蓝屏。

[#LC2277]

- 尝试使用智能卡和个人标识号 (PIN) 重新连接到 VDA 会话失败，显示下列错误消息之一：

- Insert a smart card（插入智能卡）
- Reading Card.（正在读取智能卡）

如果用户通过使用用户名和密码登录，会话将建立连接。

[#LC2596]

系统异常

- 操作系统在 picadm.sys 上遇到错误，显示蓝屏和检测错误代码 0x22。

[#LC0227]

- 在直通会话中使用媒体播放器播放视频可能会导致会话意外退出。

[#LC0553]

- Microsoft Excel 2010 的已发布版本在打开的电子表格文件之间移动时失败。

[#LC0809]

- VDA 在 ctdvcs.sys 上遇到错误，并在蓝屏上显示错误检测代码 E3。

[#LC0857]

- 如果 VDA 会话在使用图形应用程序时最大化，VDA 可能会变得无响应。

[#LC0908]

- 在下列一种情况下，Windows 资源管理器可能会异常关闭：

- 选择大量文件且文件名中包含的字符数超过 260，然后选择“发送至 > 传真收件人”选项。
- 尝试打开第三方应用程序。

- 尝试使用 Nitro PDF 合并文件。
[#LC0938]
- ICA WSK 传输驱动程序 (tdwsk.sys) 中的页面故障导致 VDA 失败并在蓝屏上显示错误。
[#LC1032]
- VDA 从版本 7.6 降级到 5.6.400 后，操作系统在 vd3dk.sys 上遇到错误并在蓝屏上显示停止代码 0x3B。
[#LC1200]
- 操作系统在 picadm.sys 上遇到错误并显示蓝屏。当“TrucationResource”的值变为 null 时会发生此问题。
[#LC1430]
- 在 ICA 会话中，从启用 SSL 的 Receiver for HTML5 播放音频文件导致 VDA 遇到致命异常，显示蓝屏。
[#LC1752]
- 有时，VDA 可能会意外关闭，然后又重新启动。
{#LC1876}
- 操作系统在 ctxdvc.sys 上遇到错误，显示蓝屏和检测错误代码 3B。
[#LC2119]
- 操作系统在 tdica.sys 上遇到错误，显示蓝屏和检测错误代码 44 [MULTIPLE_IRP_COMPLETE_REQUESTS]。
[#LC2171]
- 用户连接 Receiver for Windows 时，Windows Server 2008 R2 和 Windows Server 2012 R2 可能会遇到致命异常。
[#LC2179]
- 从版本 5.6 的 Rollup 4 升级到版本 7.6 后，VDA 在关闭时可能会变得无响应。
[#LC2245]
- 服务器可能会在 icausb.sys 上遇到错误，并在蓝屏上显示消息 PNP_DETECTED_FATAL_ERROR。
[#LC2811]
- wfshell.exe 进程可能意外关闭并阻止建立新连接。
[#LC3024]
- 重新连接到断开的会话时，组策略引擎 (CitrixCseEngine.exe) 可能会意外关闭。
[#LC3091]
- 如果 VDA 会话在使用图形应用程序时最大化，VDA 可能会变得无响应。
[#LC3279]
- VDA 可能会在 vdtw30.dll 上遇到致命异常，并显示蓝屏。
[#LC3354]

- VDA 可能会在 vd3dk.dll 上遇到致命异常，并显示蓝屏和停止代码 0x50。如果将基于 Java 的升级工具配置为在启动时升级 VDA，会出现此问题。

[#LC3372]

- 尝试连接到会话时 explorer.exe 进程可能会意外退出。

[#LC3411]

用户体验

- 此修复解决了下列关于应用程序在 Receiver for Windows 工具栏上显示方式的问题：

- 单个应用程序可以显示默认的 Receiver 图标而非应用程序的特定图标。
- 当多个应用程序运行在同一会话中时，它们可以按组显示在单个默认 Receiver 图标下。

[#LC0094]

- 此修复解决了版本 4.200 之前的 Receiver for Windows 桌面组合重定向通过 LAN 连接使用时的可用性问题。

[#LC1507]

用户界面

- 有时，当用户重新连接到 VDA 并使用多个显示器时，窗口位置不保留。

[#LC2075]

- 用户从 Receiver for Windows 或移动 Receiver 打开已发布的应用程序时，“自动显示键盘”可能不显示。

[#LC2747]

其他

- 此修复向 vdaredirector.exe 引入了 /Verbose 选项，导致在 vdaredirector.exe 运行时显示消息窗口。如果不指定 /Verbose 选项，将不会显示消息窗口。

[#LC0338]

- 在 Microsoft Windows 8 (32 位版本) VDA 上安装第三方防病毒软件后，Citrix Receiver 会话无法启动。

[#LC1636]

- 如果 VDA 上存在包含修复 # LA5736 的修补程序或下列任何修补程序，用户将无法在控制台会话中更改 Lync 状态：

- ICAWS750WX64007
- ICAWS750WX64008
- ICAWS750WX64012
- XD710ICAWSWX64003

Lync 状态始终显示为“Away”（离开）。

重要：应用此修补程序后，如果部署中存在 NetScaler Gateway，当用户从 VDA 会话断开连接，然后再连接到控制台会话时，Lync 状态仍显示为“Away”（离开）。

[#LC1691]

- 如果 VDA 上存在包含修复 # LA5736 的修补程序或下列任何修补程序，用户将无法在控制台会话中更改 Lync 状态：

- ICAWS750WX64007
- ICAWS750WX64008
- ICAWS750WX64012
- ICAWS750WX64028
- XD710ICAWSWX64003

Lync 状态始终显示为“Away”（离开）。

[#LC2479]

- 本版本支持 Framehawk 虚拟通道，这是新 ICA 虚拟通道，扩展了 Citrix HDX 技术以改进使用经常出现数据包丢失和延迟问题的宽带无线连接时的用户体验。有关详细信息，请参阅[新增功能](#)。

[#LC2782]

- 这是一项支持特定五合一 COM 端口设备的增强功能。它可确保设备的 ID 卡读卡器功能可在 VDA 会话中正常使用。

[#LC0325]

- 如果用户间接属于设置为限制能见度的嵌套组，尝试启动 VM 托管应用程序可能会失败。

[#LC3589]

XenApp 和 XenDesktop 7.6 Feature Pack 3 (7.6.300) 与 7.6 LTSR 中包含的 VDA for Server OS 的版本相同。XenApp 和 XenDesktop 7.7 中包含的版本包含的修复与版本 7.6.300 相同，并且还包含 XenApp 和 XenDesktop 7.7 的兼容性更新。

Broker Agent

HDX 3d Pro

无缝窗口

HDX MediaStream Flash 重定向

会话/连接

HDX MediaStream Windows Media 重定向

站点/场管理

安装、卸载、升级

智能卡

键盘/鼠标

系统异常

本地应用程序访问

用户体验

登录/身份验证

用户界面

打印

其他

Broker Agent

- VDA 可能会卡在注册过程的初始化状态。Citrix Desktop Service 在未经过重新启动的情况下运行数日后会发生此问题。

[#LC0570]

- 启用函数“CName”时，VDA 注册可能需要非常长的时间。应用此修复后，打开 CNameSuppressOrgFqdnLookup 值会加快注册过程的速度。

重要：如果启用了 CNameSuppressOrgFqdnLookup，则必须启用 CName。

要启用此修复，必须创建以下注册表项：

- HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\VirtualDesktopAgent
名称：UseCnameLookup
类型：REG_DWORD
值：已启用 = 1 (已禁用 = 0)
- HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\VirtualDesktopAgent
名称：CNameSuppressOrgFqdnLookup
类型：REG_DWORD
值：已启用 = 1 (已禁用 = 0)

[#LC0791]

- 在多宿主服务器中尝试启动已发布的应用程序可能会失败并显示以下错误：

“There is no Citrix XenApp server configured on the specified address.” (指定地址上未配置任何 Citrix XenApp 服务器。)

[#LC1686]

- 如果 VDA 版本 7.6 是 XenDesktop 5.6 站点的成员，当用户在连接后又断开连接，然后再重新连接时，重新连接尝试将失败，且 VDA 未注册。

[#LC1859]

HDX 3D Pro

- 应用此修复后，HDX 网络通信的带宽容量为 20 兆位/秒 (Mbps)。

[#LC2780]

HDX MediaStream Flash 重定向

- 使用 Microsoft Internet Explorer 9 或此浏览器的更高版本时，如果在另一个 Flash 重定向视频正在播放期间单击某个 Flash 重定向视频的链接，两个视频的视频流会重叠。

此修复并未消除此问题，但是通过在服务器上设置以下注册表项，可以缩短此问题持续的时间：

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer
名称：UrlListForTerminationFlashInst
类型：REG_MULTI_SZ
数据：<受影响的 Web 站点的 URL；同时包括 http:// 和 https://，每个 URL 单独占用一行；例如：
http://www.youtube.com/; https://www.youtube.com/>

[#LC0453]

- 在 Microsoft Internet Explorer 8 或此浏览器的更高版本中使用 YouTube 的内置“Search”（搜索）字段时，HDX Flash 重定向恢复为服务器端呈现。

要启用此项修复，必须在服务器上设置以下注册表项：

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer

名称：SupportedUrlHeads

类型：REG_MULTI_SZ

数据：<每个值单独占用一行，不分隔>

http://

https://

file://

注意：SupportedUrlHeads 最初是在修复 #LA4151 中引入的。此修复也使用该注册表项。如果向注册表中添加 SupportedUrlHeads，Flash 重定向将忽略 Flash 实例化请求，除非目标 URL 的开头为 SupportedUrlHeads 中指定的其中一个标头。

[#LC0505]

- 使用 Microsoft Internet Explorer 10 时，cnn.com Web 站点中嵌入视频的 Flash 重定向可能会失败。如果此 Web 站点上的第一个视频自动播放，将会出现此问题。此视频的页面上的一些内容（如标题、状态和广告）可能会缺失。

[#LC0830]

- 在下列情况下，尝试播放 Web 站点上的视频时，Internet Explorer 可能会异常关闭：

- 启用选项“禁用 Flash 加速”，或者站点位于黑名单上。
- 用户使用的是不支持 Flash 重定向的 Receiver 版本，如 Receivers for Mac OS X、iPad、iPhone 和 Android。

[#LC2256]

HDX MediaStream Windows Media 重定向

- 安装 Receiver for Windows 的 4.1 或 4.1.2 版本后，如果启用 HDX MediaStream 多媒体加速，Windows Media Player 可以加载视频，但是视频进行不到第一个帧后面的内容。

[#LC1460]

- 此版本修复了 HDX 图形编码器生成的冗余帧，实现了更为顺畅的帧速率和更好的交互性。

[#LC2177]

- 在使用 Receiver for Windows 13.3 或 13.4 版本的 VDA 会话中播放视频文件时，自定义应用程序中出现音频和视频同步。但是，升级到更高版本的 Receiver 后，Windows Media 重定向失败。出现此问题后，用户只能听到声音或只能看到视频。

要启用此修复，必须使用 Citrix Receiver for Windows 4.3。

[#LC2516]

- 尝试关闭其中的视频已暂停的 Windows Media Player 会导致 Windows Media Player 无响应。

[#LC3410]

安装、卸载、升级

- 如果在安装通用打印服务器客户端后安装 Hotfix Rollup Packs 1 至 4 中的任何修补程序，会话打印机枚举可能需要花费相当长

的时间。

[#LA5977]

- 安装 2014 年 9 月之前发布的适用于 Windows 桌面操作系统和服务器操作系统的 XenApp 7.5、XenDesktop 7.1 和 7.5 VDA Core Services 的修补程序，会导致 ICA 会话性能监视器计数器被删除。这会对倚赖这些计数器的工具和进程的运行产生不利影响。此修复恢复了因为安装早期修补程序而被删除的计数器，并阻止在安装后续修补程序时将其删除。

[#LC0771]

- 安装修补程序 XA650R04W2K8R2X64008 之后，在早期 XenApp 版本上可以正常使用的现有图标可能会报告为损坏。

[#LC1081]

- 安装包含修复 #LA5872 的修补程序或 Hotfix Rollup Pack 之后，用户登录 Receiver 时，Windows 登录安全通知直到用户已发布的应用程序启动后才显示。

[#LC1215]

- 安装 XA650W2K8R2X64R04 之后，Discovery 有时会失败，并且可能需要尝试多次才能成功运行 Discovery。

[#LC1369]

- 安装修补程序 XA650W2K8R2X64R03、XA650W2K8R2X64R04 或 XA650W2K8R2X64R05 之后，可能会出现以下问题：

- 当以下注册表项使用德文二进制时，事件查看器失败：
 - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\WINEVT\Publishers\{c6d0a271-2e59-40cd-b451-9000a9f104f9}
 - MessageFileName : C:\Program Files (x86)\Citrix\System32\resource\de\RaveEventMessages.Dll
 - ResourceFileName : C:\Program Files (x86)\Citrix\System32\resource\de\RaveEventMessages.Dll
- 如果操作系统使用英文，默认用户帐户设置为使用德文，则 RAVE 事件查看器将使用德文而非英文。

[#LC2130]

- 安装 Hotfix Rollup Pack 5 之后，XTE 服务失败并放弃用户会话。

{#LC2180

- 将 VDA 从 XenDesktop 7.5 升级到 XenDesktop 7.6 后，用户尝试进行连接时，显示错误 1030。系统事件日志中显示以下消息：“The Citrix ICA Transport Driver received an invalid Transport packet on port 2598.”（Citrix ICA 传输驱动程序在端口 2598 上收到无效的传输数据包。）

[#LC2501]

- 在 XML Broker 上安装 XA650W2K8R2X64R05 之后，XML Service 对用户进行身份验证所花费的时间会增加。在负载过大的环境中，此时间增加可能过长，从而导致 Web Interface 的用户会话过时。

[#LC2536]

键盘/鼠标

- 在同时使用 Receiver for Linux 和 Czech 键盘布局的会话中，某些键可能无法正常使用。以下字符可能会出现错误映射：
 - d' 错误地映射为 ě
 - ñ 错误地映射为 ň
 - ě 错误地映射为 ř

注意：安装此修复后，从 Receiver for Linux 会话内打开 RDP 会话时，Shift 键可能会间歇性出现问题。要阻止此问题，Citrix 建议在 Windows 服务器上安装 Microsoft 文章 [KB2592687](#)。此外，请参阅知识中心文章 [CTX110281](#) 以了解解决方法。

[#LA3968]

- 通过使用鼠标在本地应用程序与已发布的应用程序之间切换时，键盘输入不起作用。

[#LA5849]

- 鼠标指针可能无法识别桌面的边界。

[#LC0383]

- 可能会在会话内间歇性丢失鼠标输入。

[#LC0632]

- 如果配置了“自动显示键盘”策略以显示用户设备上的键盘，键盘不会自动显示在已发布的应用程序中。

注意：要完全修复此问题，请在用户设备上安装修复 #LC3158。

[#LC2432]

- 在 iPad 设备上使用 OpenOffice 应用程序时，自动显示键盘功能不起作用。

[#LC2717]

本地应用程序访问

- 在全屏模式的桌面会话中，如果启用本地应用程序访问且禁用“拖动时显示窗口内容”设置，在 Internet Explorer 中的图片上方移动应用程序图标时，高分辨率图片会变形。

[#LC1722]

登录/身份验证

- 用户尝试通过使用智能卡身份验证的 StoreFront 启动应用程序时，显示错误消息“未找到有效证书”。这会阻止用户登录。

[#LC1704]

打印

- Citrix Print Manager Service (CpSvc.exe) 进程可能会意外退出。

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

- 漫游到不同用户设备时，可能会将错误的打印机设置为默认会话打印机。

[#LC0225]

- 此修复解决了阻止将以下打印机映射到 Receiver for Linux 用户会话中的问题：

- Epson-LQ-1050
- Epson PLQ-20

要启用此项修复，必须设置以下注册表项：

HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\ICA\PrintingSettings

名称：AllowPrinterMappingByName

类型：REG_DWORD

数据：1

[#LC0389]

- 重新连接到已断开连接的延迟会话，并且已发布的应用程序采用指定的窗口大小时，会话打印机可能无法映射到会话。

[#LC0615]

- Citrix Print Manager Service (CpSvc.exe) 由于访问冲突而失败。

[#LC0660]

- 如果启用“会话延迟”，重新连接到会话时，用户设备上的重定向打印机可能不会进行映射。

[#LC0816]

- 此修复恢复了“旧版打印机名称”策略，此策略之前在该产品的 7.x 版本中不可用。

[#LC1002]

- 用户登录新会话时，不显示打印机列表。当用户重新连接到已断开连接的会话时，不显示打印机列表。

[#LC1226]

- 启用了“自动创建一般通用打印机”策略设置时，Citrix Print Manager Service (CpSvc.exe) 将不创建 Citrix 通用打印机一般打印机。

[#LC1255]

- 此功能增强了通用打印服务器和通用打印客户端的实现性能计数器。

[#LC1820]

- 尝试通过使用 Services 管理单元重新启动 Print Manager Service 失败。显示错误消息“Couldn't stop this service”（无法停止此服务），且 Print Manager Service 的状态显示为“stopping”（正在停止）。

[#LC2122]

- 即使将“通用打印 EMF 处理模式”策略设置成“为打印机重新处理 EMF”，此策略仍保留默认值设置。

[#LC2521]

- Citrix Printer Manager Service (Cpsvc.exe) 异常停止并显示访问冲突错误。

[#LC2624]

- 用户尝试打印文档或选择其他打印机时，可能会遇到间断性的延迟，并在长达一分钟的时间内看到消息“Connecting to printer”（正在连接到打印机）。

[#LC2701]

- 使用 Citrix 通用打印机驱动程序时，打印的第一页可能为空白。

[#LC2771]

- 用户通过 Receiver 登录时，无法使用 Diebold 打印机（型号 P361）进行打印。

[#LC2908]

- 此增强功能会在用户登录时为每个端口创建一个自动还原的打印机。

[#LC2943]

无缝窗口

- Microsoft Excel 已发布的无缝实例可能会遇到屏幕闪烁。本地运行的应用程序可能会丢失键盘焦点。

安装此修复后，必须创建以下注册表项，然后创建以下两个值：

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\wfshell\TWI\XLMAIN

名称：ClassName

类型：REG_SZ

数据：XLMAIN

名称：Type

类型：REG_DWORD

数据：0x00000808

[#LC0401]

- 移动某个停靠应用（如 Windows 任务栏）时，最大化的无缝应用程序不会正确调整大小。

[#LC1342]

- 具有提升权限且在 Citrix Studio 中发布的应用程序无法采用无缝模式启动。

[#LC1537]

- 在无缝会话中，当已发布工具栏的无缝标识设置为“开”时，尝试在辅助显示器中停靠无缝工具栏总是导致停靠在主显示器上。有关无缝标识的详细信息，请参阅 Citrix 文章 [CTX101644](#)。

[#LC1599]

- 尽管应用程序进程正在运行且服务器 VDA 上存在会话，无缝窗口仍不显示。

[#LC1728]

- 在多显示器环境中，如果用户将无缝窗口移动到辅助显示器上，然后更改主显示器或辅助显示器的分辨率，无缝窗口可能会移动到主显示器上。

[#LC1819]

会话/连接

- 用户尝试通过客户端驱动器映射将某个文件保存到用户设备上时，保存到映射驱动器上的文件可能不包含任何数据。

[#LA3566]

- 将文件复制到映射的客户端驱动器上后，可能会错误地计算并显示映射驱动器上的剩余磁盘空间。

[#LA4799]

- 尝试重新连接到用户可能会遇到长达 60 秒的登录延迟，即使通过 GPO 配置了多个 WMI 过滤器也是如此。

要禁止在重新连接时进行策略计算，必须设置以下注册表项：

HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\Reconnect

名称：DisableGPCalculation

类型：REG_DWORD

数据：1

[#LA4932]

- 系统可能不会在重新连接时重新计算 ConnectTime。 结果导致 EdgeSight 报告中某些会话的 Session Creation Server Duration (SCSD) 和 Session Startup Duration (SSSD) 值可能异常高。

要启用此项修复，必须设置以下注册表项：

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\wfshell

名称：DisableServerStartupEventForReconnect

类型：REG_DWORD

值：1

[#LA5141]

- 用户使用 Receiver for Mac 连接时，尝试播放 Microsoft Outlook 中的音频文件可能失败。

[#LA5657]

- 启用 Excelhook 时，尝试在工作簿之间切换可能失败，Excel 可能出现无响应的情况。

[#LA5708]

- 重新启动 XenApp 服务器后，尝试连接到许可证服务器可能失败。

要启用此修复，必须创建以下注册表项：

- 在 32 位 Windows 上：

HKEY_LOCAL_MACHINE\Software\Citrix\Licensing

名称：RA_retries

类型：REG_DWORD

值：1（如果值不存在或小于 5，则默认值视为 5）

- 在 64 位 Windows 上：

HKEY_LOCAL_MACHINE\Software\Wow6432Node\Citrix\Licensing

名称：RA_retries

类型：REG_DWORD

值：1（如果值不存在或小于 5，则默认值视为 5）

[#LA5773]

- 尝试重新连接锁定的已发布桌面会话可能会失败，并出现以下错误消息：

“连接错误：Citrix 服务器已达到此应用程序的并发应用程序限制。 请与系统管理员联系。”

出现此问题时，除了现有会话，还会启动新会话。

[#LC0158]

- 尝试重新连接到已断开连接的会话可能失败，并启动一个新会话而非原始会话。启用“Restrict Remote Desktop Services users to a single remote session”（限制远程桌面服务用户只能连接单个远程会话）设置时会出现此问题。

[#LC0201]

- 与 VDA 建立远程桌面 (RDP) 连接，然后从 RDP 会话断开连接时，事件日志中显示以下错误消息（事件 ID 1048）：

Citrix Desktop Service 正在重新注册 DDC:

NotificationManager:NotificationServiceThread: WCF 故障或被 Broker 拒绝 (DDC: bruin.newer.In)

[#LC0251]

- 将应用程序实例限制设置为“1”时，尝试从已断开连接的会话重新启动应用程序可能失败，可能会启动新会话。客户端自动重新连接尝试为内部会话重新连接状态设置错误的值时会导致此问题。

[#LC0259]

- 打开已发布的应用程序，然后运行 explorer.exe 命令以打开 Windows 资源管理器会导致打开桌面而非 Windows 资源管理器。

注意：安装此修复后，explorer.exe 进程可能会意外退出。有关详细信息，请参阅知识中心文章 [CTX128009](#)。

[#LC0285]

- 在两个映射驱动器的 UNC 路径之间复制文件时，如果第二个驱动器是通过使用 Windows Substitute 命令 (SUBST) 从第一个 UNC 驱动器创建的，文件可能会损坏。

[#LC0288]

- 启用“多流”策略时，剪贴板数据从服务器传输到用户设备可能需要很长时间。

您可以创建以下注册表项，以便将默认服务器输出带宽更改为较大的值：

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\Wds\icawd

名称：InitialOutputSpeed

类型：REG_DWORD

值：10000 – 1000000（10 KB/秒 - 1000 KB/秒；建议值：300000）

[#LC0390]

- 按照知识中心文章 [CTX133198](#) 中所述配置 Excelhook 后，在 Microsoft Excel 和 Microsoft VB for Applications 均在运行的会话中，会发生以下问题：

- 即使 VB for Applications 窗口已获得焦点，主 Excel 窗口仍保留在前台。
- 关闭 VB for Applications 窗口也会关闭主 Excel 会话。

[#LC0398]

- 用户会话可能会在注销时停留在断开连接状态。结果导致会话无法关闭，您也无法重新启动服务器。

[#LC0399]

- 尝试在 XenApp 中发布 Mozilla Firefox 28 及更高版本失败并显示以下错误消息：

“System.NotSupportedException: 数据的格式不是 ICO”

[#LC0402]

- Windows Server 2008 R2 (服务器操作系统) 上运行的 Citrix XenApp 7.6 和 XenDesktop 7.6 VDA Core Services 可能会在显示“欢迎”屏幕时变得无响应。如果发生此问题，与服务器建立新的 Receiver 和远程桌面 (RDP) 连接将失败。

[#LC0405]

- 有时，用户会话可能会变得无响应，并且无法注销。如果用户手动从控制台注销，也会发生此问题。

[#SLC0472]

- 在已发布的桌面上，用户登录或启动应用程序时，状态指示器进度条可能无法完成，并且可能无法显示某些登录状态消息。

[#LC0482]

- 在 Microsoft Silverlight Web 应用程序中的屏幕转换期间，移动设备（如 iOS 和 Android）上的“自动显示键盘”意外显示。

[#LC0499]

- 用户从任务栏关闭 Visual Basic Editor 的已发布版本时，打开的已发布 Excel 电子表格也关闭。

[#LC0534]

- 启动已发布的应用程序时，用户可能会遇到登录缓慢的情况。

[#LC0596]

- 如果在 XenApp 6.5 中启用“移动体验”策略，用户连接到 Receiver for iOS 后，注销时会显示以下错误消息：

“There was a problem logging off. Please contact the help desk for assistance.”（注销时遇到问题。请与技术支持人员联系以获得帮助。）

[#LC0628]

- 如果用户密码包含的字符数超过 45 个，尝试连接到运行 Windows Server 2008 或 Windows Server 2012 VDA 的 XenDesktop 7.x 可能会失败。

[#LC0689]

- 如果启用“允许时区重定向”选项，在运行于非英文 Windows 7 用户设备上的会话中，时区同步可能无法用于已发布的 Java 应用程序。出现此问题是因为 GetDynamicTimeZoneInformation API 无法向 Receiver 会话返回时区信息。

在国际语言用户设备上，将系统区域设置设置为与语言包相同的语言。

注意：此修复无法解决俄语时区的问题。

[#LC0807]

- 启用客户端驱动器映射时，有时会向已发布的 VB 应用程序使用 FileStream() 函数写入的文件末尾附加意外字符。

[#LC0815]

- 在用户设备上，尝试远程“区域和语言 > 日期和时间格式”下面的组合框控件可能会失败。如果组合框控件已经被成功远程过一次，第二次尝试远程此控件会发生此问题。

[#LC0836]

- 用户尝试连接到已断开连接的会话时，建立连接需要长达三分钟时间。

[#LC0850]

- 启动已发布的应用程序时，用户设备上会出现以下错误消息：

“The Citrix server is unable to process your request to start this published application at this time. Please try again later. If the problem persists, contact your administrator.” (Citrix 服务器此时无法处理您的请求以启动此已发布的应用程序。请稍后重试。如果问题仍然存在，请联系管理员。)

解决方法：wfshell 进程在启动应用程序前需要一定时间才能完成对应用程序属性的检查。根据具体的网络条件，请求可能会在检查完成前超时，同时显示错误消息。

此修复引入了对以下注册表项的支持，此注册表项允许您将应用程序属性检查的超时时间最高延长至 4 分钟：

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\ICAST

名称：WaitTimeOutValue

类型：REG_DWORD

值：<所需的超时延迟，单位为毫秒。默认值：12000；支持的最大值：240000>

[#LC0860]

- 用户登录适用于移动设备的任何 Receiver 版本，然后再打开启用“自动显示键盘”策略的已发布 Lotus Notes 版本时，在“收件人”、“抄送”和“密件抄送”字段中输入电子邮件地址时，键盘不显示。

[#LC0899]

- VDA 变得无响应时，从 Receiver 显示“Connection Established, Negotiating Capabilities”（已建立连接，正在协商功能）弹出窗口。要成功启动任何其他会话，需要重新启动 VDA。

[#LC0909]

- 用户使用 Receiver for Windows 登录时，时区不正确。要使此修补程序正常运行，必须安装以下修补程序：

- 在用户设备和服务器上安装相同的 Microsoft 时区更新修补程序。例如，如果用户设备上安装了 Microsoft 修补程序 [KB2998527](#)，服务器上也应安装此修补程序。
- 如果服务器操作系统为 Windows Server 2008 R2 Service Pack 1，在此服务器上安装 Microsoft 修补程序 [KB2870165](#)。
- 在用户设备上安装修复 #LC1392。

[#LC1061]

- 尝试重新连接到服务器上的活动会话可能会失败。将应用程序实例限制设置为“1”时，尝试打开其他会话可能会失败，用户必须注销或从现有会话断开连接才能重新连接或启动新会话。

[#LC1084]

- 如果在属于场的服务器关闭期间域之间的信任关系发生改变，尝试枚举应用程序可能会失败。尝试使用 UPN 进行登录成功，但是尝试使用下级登录名失败。

[#LC1136]

- 如果 USB 重定向策略包含的字符数超过 1000，所有 USB 驱动器都会重定向，即使存在设备的拒绝规则也是如此。

[#LC1153]

- 如果在 Microsoft Project 2013 中启用 ExcelHook 并且打开多个项目，打开的第一个项目可能不会显示在任务栏中。

[#LC1158]

- 有时，无缝应用程序的视觉设置（如标题字体、滚动条大小和菜单高度等）可能无法与用户设备设置正确同步。

[#LC1169]

- 会话共享可能会间歇性失败。

[#LC1184]

- 如果另一个进程与 picadmsys 占用相同的锁，用户无法从会话注销，会话保持在断开连接状态。

[#LC1243]

- 如果在 Desktop Delivery Controller (DDC) 上启用“设为无损”和“允许视觉无损压缩”设置，最终图像质量可能会降级。

[#LC1271]

- 用户首次登录时，在 Receiver 会话中屏幕可能会显示已损坏。将视觉质量更改为“设为无损”并启用“允许视觉无损压缩”设置时会出现此问题。

[#LC1272]

- 尝试在 Microsoft Access 2007 的已发布版本中复制信息失败。此信息不显示在 Office 剪贴板中。

[#LC1290]

- Microsoft Excel 2010 的已发布版本在打开的电子表格文件之间移动时失败。

[#LC1356]

- 在 XenDesktop 和 XenApp 会话中，重定向可能无法正确用于 MailTo 功能。用户单击 Web 浏览器中的“MailTo”链接时，不是打开已发布的 Microsoft Outlook，而是显示错误。

[#LC1400]

- 启用会话延迟并在 iOS 或 Android 设备上关闭某个应用程序时，显示黑色窗口，且此窗口保留在设备屏幕上。

[#LC14074]

- 用户使用客户端自动重新连接功能重新连接到会话时，客户端时区可能无法在用户设备中重定向，相反，服务器端时区可能会显示在用户设备上。

[#LC1426]

- 用户重新连接到其桌面会话时，“Session Creation Server Duration (SCSD)”和“Session Startup Server Duration (SSSD)”的值被重置。

[#LC1464]

- 在基于德语的 Windows 设备上，适用于德语的 Multimedia Rave (RaveEventMessages.dll) Service 可能会使控制事件日志服务的 svchost.exe 进程失败。出现此问题时，用户无法登录。

[#LC1543]

- 用户启动已发布的应用程序时，Citrix Stack Control Service (SCService64.exe) 停止响应。

[#LC1545]

- 用户从桌面会话注销时，WFShellWindow 停止响应，且消息“Wfshell needs to close”（Wfshell 需要关闭）显示三分钟。

[#LC1591]

- 在多显示器环境中，应用程序可能会在已发布的无缝工具栏应用程序后面打开。

要启用此修复，可能需要安装修复 #LC1342 和修复 #LC0491。

[#LC1673]

- 登录基于 Windows 的设备，然后连接到 XenApp 虚拟机 (VM) 时，TWI 模块 (TWI3.dll) 增加了 CPU 使用率。

[#LC1810]

- 在两个双显示器用户设备设置中，如果启动了多个具有 HDX 功能的应用程序，其中的一些应用程序可能会显示为黑色窗口。

[#LC1818]

- 录制禁用旧图形模式的 XenApp 会话时，录制文件无法回滚。

[#LC1850]

- 延迟会话中可能会出现超时，导致会话异常停止，从而显示黑色窗口。

[#LC1903]

- 用户尝试连接到已断开连接的会话时，建立连接需要长达三分钟时间。

[#LC1908]

- 如果将 Active Directory 服务器设置为只读服务器，用户无法启动应用程序，并且无限期显示消息“Connection Established, Negotiating Capabilities”（已建立连接，正在协商功能）。

[#LC1964]

- 在会话中启用“客户端自动重新连接”策略时，可能不显示身份验证提示。

要启用此修复，请设置以下注册表项：

- HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\WFShell
名称：SessionReconnectMinTimeInMilliseconds
类型：REG_DWORD
值：最大会话长度
- HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\WFShell
名称：LockOnReconnect
类型：REG_DWORD
值：1

配置注册表项后，请重新启动服务器或运行命令“gpupdate /force”。

[#LC1990]

- 启动已发布的应用程序时，Citrix Graphics (CtxGfx.exe) 进程导致 CPU 使用率高于正常值。

[#LC2007]

- 当用户使用 Citrix Receiver for Windows 从会话断开连接，然后在屏幕分辨率更小的设备上使用 Citrix Receiver for Linux 或 Citrix Receiver for Windows CE 重新连接到相同会话时，会话屏幕分辨率可能无法调整大小。如果会话启用了“旧图形模式”策略设置，可能会发生此问题。

[#LC2079]

- 用户复制从 SAP 内打开的嵌入式 Word 文件的内容时，无法将内容粘贴到新的嵌入式 Word 文件中。

[#LC2092]

- 用户从 Web Interface 启动已发布的桌面时，会话在“正在设置以下对象的个性化设置: Microsoft Windows Media Player”时变得无响应，音频无法从桌面运行。

[#LC2103]

- 如果在装有 Receiver for iOS 的 iPad 上启动 Lotus Notes，此应用程序可能会意外关闭。

[#LC2124]

- 由于将会话属性“AllowReconnect”设置为“False”，尝试重新连接到已断开连接的会话有时会失败。

[#LC2174]

- 用户从会话断开连接时，Citrix WinFrame Shell (WFSHELL.exe) 导致较高的 CPU 使用率。

[#LC2235]

- 使用 /appve 参数作为已发布的应用程序启动的 Internet Explorer 无法加载关联的数据包。

[#LC2292]

- 用户从会话断开连接后，Wfshell.exe 继续占用大量 CPU 资源。

[#LC2391]

- 使用 AutoCAD 和 AutoCAD Civil 3d 时，鼠标指针永久显示为沙漏。

[#LC2438]

- 从客户端驱动器映射 (CDM) 打开已发布的应用程序文件，然后在输入文本后保存此文件时，修改的时间戳未更新。

[#LC2305]

- 尝试处理驻留在客户端驱动器映射 (CDM) 驱动器上的文件可能导致 picadm.sys 上出现死锁，并可能导致已发布的桌面变得无响应。

[#LC2312]

- 如果 Excel.csv 文件包含大量条目，将此文件从用户设备复制到服务器可能需要很长时间。

[#LC2366]

- 最大化 Excel 或 Internet Explorer 的已发布版本时，名称框、编辑栏或菜单栏的顶部显示为白色条。

要为 Excel 和 Internet Explorer 启用此修复，请创建以下注册表项：

Excel

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\wfshell\TWI\XLMAIN

名称：ClassName

类型：REG_SZ

值：XLMAIN

名称：Type

类型：REG_DWORD

值：0x00020000

Internet Explorer

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\wfshell\TWI\IEFrame

名称：ClassName

类型：REG_SZ

值：IEFrame

名称：Type

类型：REG_DWORD

值：0x00020000

要向多个服务器应用此修复，请创建注册表标志。有关详细信息，请参阅 Citrix 支持站点上 [Seamless_Configuration.pdf](#) 中的“Seamless Per Application Window Registry Flags”。

[#LC2504]

- 启用 Excelhook 时，在打开的 Excel 工作簿之间切换可能会导致选定的工作簿丢失焦点。此外，用户无法选择任何已打开的 Excel 工作簿，焦点保留在选择的上一个文件上。

#LC2589]

- 如果启用“仅允许每个用户运行一个应用程序实例”设置或“客户端自动重新连接”策略，用户会收到错误“已打开了此应用程序的一个实例，不允许运行多个实例。请与系统管理员联系。”出现此问题是由于无法在两个林之间进行会话共享。

[#LC2592]

- 在 Windows 7 64 位计算机上 Internet Explorer 8 的已发布版本中播放时，视频在应用程序“Unreal Streaming Media Playe”中为空白。

[#LC2650]

- 通过使用客户端驱动器映射复制超过 1000 个文件可能会导致性能问题。

[#LC2702]

- 更新密码注册表项之后，用户无法启动应用程序。要应用此修复，需要安装新版本的 SSL SDK。

[#LC2729]

- 会话可能无法启动并可能显示以下错误消息：

“A device attached to the system is not functioning.”（连接到系统的设备当前未运行。）

在使用 Kerberos 保护的具有双向信任域的环境中，当用户和服务器位于不同的域中时可能出现此问题。

[#LC2762]

- 在 Windows Phone 8.1 上打开 Internet Explorer 的已发布版本时，如果将设备转为水平位置，半个屏幕会显示黑色窗口。此外，用户还无法最大化 Internet Explorer 窗口。如果用户将设备转为垂直状态，屏幕显示正常。

[#LC2786]

- 如果启用会话共享和会话延迟，断开连接的会话可能无法重新连接到同一 VDA，而是重新连接到其他 VDA。出现此问题时，VDA 上会记录事件 ID 1048、事件 ID 1050 和事件 ID 7。

[#LC2860]

- 如果已发布的应用程序停止运行，桌面窗口管理器可能不允许用户关闭应用程序。

[#LC2922]

- 用户连接到服务器上的会话时，会话可能无法进行到欢迎屏幕后面的部分。在服务器重新启动之前，服务器上的所有会话都可能发生此问题。

[#LC2936]

- 用户每次登录 VDA 时，VDA 上注册表位置 HKEY_CURRENT_USER\Control Panel\Desktop\UserPreferencesMask 中的注册表项和值可能被 wfshell.exe 进程重写。要阻止此问题，请在 VDA 上创建以下注册表项，并将值设置为 1：

HKLM\SYSTEM\CurrentControlSet\Control\Citrix

DWORD 值：

EnableVisualEffect

[#LC2965]

- 用户再次登录 VDA 时，Windows 资源管理器收藏夹菜单 %userprofile%\Links 下面已删除的桌面和下载内容快捷方式重新被创建。

[#LC2998]

- 用户重复登录和注销时，Citrix WinFrame Shell (Wfshell.exe) 停止响应。

[#LC3003]

- 用户登录 Receiver 4.2 时，显示在会话中的时间与用户设备上的时间不匹配。

[#LC3067]

- 管理员尝试启动从 System32 文件夹发布的应用程序可能会失败。

[#LC3148]

- 结合使用外部显示器和便携式计算机时，外部显示器的边角可能会显示白色正方形。当外部显示器的屏幕分辨率与主显示器的分辨率不同，且位于主显示器的左侧，其顶部边缘在主显示器的顶部边缘上方时，尤其会出现此问题。

[#LC3186]

- 通过使用负载评估程序将满载服务器置于脱机状态后，尝试连接到断开连接的会话可能会失败。

[#LC3226]

- 用户从 Receiver 4.2 连接到已发布的桌面时，Citrix WinFrame Shell (wfshell.exe) 失败。

[#LC3227]

- 在 XenApp 服务器上启动已发布的应用程序或桌面时，单点登录可能无法使用。

[#LC3301]

- 禁用 USB 重定向的情况下，从已发布的应用程序断开连接可能会导致事件查看器中显示以下虚假事件 261：

“The Citrix Device Redirector service could not complete an IO operation with the Redirector Bus.” (Citrix Device Redirector Service 无法使用重定向程序总线完成 IO 操作。)

[#LC3439]

- 退出 64bit ThinAPP 打包的应用程序 (BMC Control-M) 时，应用程序可能会在 sfrhook64.dll 上遇到意外异常。

[#LC3484]

- 有时，重新连接到断开连接的会话后，剪贴板重定向失败。如果剪贴板虚拟通道在重新连接时异常关闭，可能会发生此问题。

[#LC3511]

- SSL 连接 (HTTPS) 可能无法在 VDA 中正常使用。

[#LC3535]

站点/场管理

- 如果启用注册表值 HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\Logon\DisableStatus (值=1)，配置文件加载服务器持续时间和登录脚本执行服务器持续时间的最终用户体验监控 (End User Experience Monitoring, EUEM) 指标可能缺失。

[#LA5764]

- 尝试检查负载评估程序设置可能失败并出现以下错误：

System.Runtime.InteropServices.COMException (0x80004005): 对 COM 组件的调用返回了错误 HRESULT E_FAIL。

[#LC0359]

- 多个 XenApp 服务器在同一时间重新启动，而不是按照“重新启动计划随机时间间隔”策略中定义的那样在不同时间重新启动。

[#LC1402]

- 指定了远程桌面网关时，非管理用户无法与 XenApp 7.6 服务器建立 RDP 会话。

[#LC2371]

- Microsoft DsBind() API 函数失败后，可能会对服务器的本地主机缓存进行虚假更新。此问题的症状包括控制器上本地主机缓存的大小快速增加，以及指向 Active Directory OU 的工作进程组的内容发生更改。

[#LC2631]

- “dscheck /full workergroup”命令在验证不包含“场服务器”对象的工作组时错误地报告出错，并显示以下消息：

“Error:: Failed to retrieve Server UIDs for the WorkGroupUID.” (错误: 无法通过检索服务器 UID 获得 WorkGroupUID。)

此修复将此情况中显示的消息修改为：

“This workergroup is configured for Farm Server and has no Server UID, verification OK.” (此工作组是为场服务器配置的，没有服务器 UID，验证无误)

[#LC3383]

- 重新启动计划随机时间间隔设置对使用 Provisioning Services 置备的服务器不起作用。

[#LC3430]

- 应用程序发布到工作组时，运行命令 dscheck /full apps /servercheck 错误地将工作组对象作为未知服务器返回，并提供将其删除的选项。

[#LC3556]

智能卡

- Citrix Smart Card Service 可能会导致 CPU 使用率非常高。要纠正此问题，请重新启动 Citrix Smart Card Service。

[#LA5919]

- 升级 VDA 后，用户尝试使用智能卡登录时，Microsoft Outlook 不会提示输入个人标识号 (PIN)，而是会显示以下错误消息：

“请插入智能卡。”

[#LC0556]

- 通过使用智能卡登录服务器后，服务器可能会在显示“欢迎”屏幕时变得无响应，并拒绝接受新会话。重新启动 Smart Card Service 可解决问题，服务器将继续接受新会话。

如果已经安装了修复 #LA0983 并在安装上一个修补程序期间配置了注册表设置，可能无须配置以下注册表项：

HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\SmartCard

名称：TransactionTimeoutEnable

类型：REG_DWORD

值：1（启用）

HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\SmartCard

名称：TransactionTimeoutValue

类型：REG_DWORD

值：<大于 5 秒的任何值>

HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\SmartCard

名称：SendRecvTimeout

类型：REG_DWORD

值：最大超时值，单位为秒；应为 30 秒或更大的值。默认将任何低于此值的值视为 30 秒。该值必须至少比“TransactionTimeoutValue”大 10 秒。

[#LC0910]

- 连接到端点的读卡器中存在智能卡时，使用 Citrix Receiver for Linux 进行身份验证后，会话可能会无响应。

[#LC0982]

- ActivIdentity (acompkcs.exe) 进程可能会意外退出并在 scardhook64.dll 上出现异常。结果导致需要智能卡的 Web 站点无法正常运行，直到重新启动 VDA 后才可以。

[#LC1056]

- Citrix Smart Card Service 可能变得无响应，并且在重新启动此服务前用户可能无法登录。

[#LC1276]

- 在会话登录期间从读卡器删除智能卡可能会阻止会话在服务器上正常终止。结果，服务器可能会进入阻止任何后续会话进行智能卡登录的状态。

[#LC3020]

- 使用智能卡连接到已发布的桌面时，Citrix Smart Card Service 可能变得无响应。

[#LC3421]

系统异常

- 操作系统在 picasers.sys 上遇到错误，显示蓝屏和检测错误代码 0x0000000A。

[#LC0222]

- 操作系统在 picadms.sys 上遇到错误，显示蓝屏和检测错误代码 0x22。

[#LC0227]

- 在已安装 Hotfix Rollup Pack 3 或 4 的服务器上，远程桌面服务在充当 XML Broker 的 XenApp 6.5 服务器上可能会意外退出。

[#LC0440]

- 服务器可能会在 vdtw30.dll 上遇到致命异常，显示蓝屏。

[#LC0471]

- 操作系统在 picadms.sys 上遇到错误，显示蓝屏和检测错误代码 0x00000022。

[#LC0483]

- 在已安装修复 #LA5282（最初包含在 XA650W2K8R2X64R04 - XenApp 6.5 Hotfix Rollup Pack 4 中）的系统上的会话中，单击已发布应用程序中的 URL 可能导致 winlogon.exe 进程意外退出。结果，已发布的应用程序断开连接。

[#LC0523]

- Citrix XML Service 可能意外失败。

[#LC0548]

- 在直通会话中使用媒体播放器播放视频可能会导致会话意外退出。

[#LC0553]

- 服务器可能会在 vdtw30.dll 上遇到致命异常，并在蓝屏上显示检测错误代码 0x0000007E。

[#LC0599]

- 操作系统在 picadm.sys 上遇到错误并显示蓝屏。

[#LC0623]

- 服务器可能会在 picadm.sys 上遇到错误，显示蓝屏和检测错误代码 0x0000000A。

[#LC0649]

- 从远程桌面 (RDP) 会话注销时，winlogon.exe 进程可能会在 twi3.dll 上意外退出。

[#LC0793]

- VDA 可能会在 picadd.sys 上遇到错误并显示蓝屏。

[#LC0798、#LC1096]

- VDA 在 ctdvcs.sys 上遇到错误，并在蓝屏上显示错误检测代码 E3。

[#LC0857]

- 操作系统在 picadm.sys 上遇到错误，显示蓝屏和停止代码 0x0000003B。

[#LC0896]

- 在下列一种情况下，Windows 资源管理器可能会异常关闭：

- 选择大量文件且文件名中包含的字符数超过 260，然后选择“发送至 > 传真收件人”选项。
- 尝试打开第三方应用程序。
- 尝试使用 Nitro PDF 合并文件。

[#LC0938]

- ICA WSK 传输驱动程序 (tdwsk.sys) 中的页面故障导致 VDA 失败并在蓝屏上显示错误。

[#LC1032]

- 启动已发布的应用程序或桌面时，如果 Citrix Audio Redirection Service 无法启动，Windows 登录用户界面 (LogonUI.exe) 或 explorer.exe 进程可能会在 icaendpoint.dll 上意外退出。

[#LC1167]

- 操作系统在 picadm.sys 上遇到错误并显示蓝屏。

[#LC1239]

- 服务器可能会在 vdtw30.dll 上意外退出。

[#LC1328]

- 服务器可能会在 vdtw30.dll 上遇到致命异常，显示蓝屏。

[#LC1414]

- twi3.dll 上的一个错误可能导致服务器停止响应。

[#LC1417]

- 终端服务的 svchost.exe 进程可能会异常关闭，应用程序事件 ID 为 1000，并导致到服务器的远程桌面和 Receiver 连接失败。

[#LC1419]

- 操作系统在 picadm.sys 上遇到错误并显示蓝屏。当“TrucationResource”的值变为 null 时会发生此问题。

[#LC1430]

- vdtw30.dll 上的错误导致蓝屏上显示检测错误代码 0x0000003B, {c0000005, fffff96000863133, fffff880099aaf70, 0}。

[#LC1567]

- 服务器可能会在 picadm.sys 上遇到死锁，阻止服务器接受新用户连接。

[#LC1685]

- 如果存在无效会话密钥，Citrix Stack Control Service 意外退出。

[#LC1717]

- 在 ICA 会话中，从启用 SSL 的 Receiver for HTML5 播放音频文件导致 VDA 遇到致命异常，显示蓝屏。

[#LC1752]

- svchost.exe 进程可能会由于 ICA 音频终结点 dll (icaendpoint.dll) 而意外停止。

[#LC1775]

- 有时，VDA 可能会意外关闭，然后又重新启动。

{#LC1876}

- 终端服务可能会由于 RPM.dll 而意外关闭。

[#LC2031]

- 服务器可能会在 vdtw30.dll 上遇到致命异常，并在蓝屏上显示检测错误代码 0x0000003B。

[#LC2037]

- 操作系统在 ctxdvc.sys 上遇到错误，显示蓝屏和检测错误代码 3B。

[#LC2119]

- 操作系统在 tdica.sys 上遇到错误，显示蓝屏和检测错误代码 44 [MULTIPLE_IRP_COMPLETE_REQUESTS]。

[#LC2171]

- 用户连接 Receiver for Windows 时，Windows Server 2008 R2 和 Windows Server 2012 R2 可能会遇到致命异常。

[#LC2179]

- picadm.sys 上出现错误，并在蓝屏上显示错误检测代码 0x00000022。

[#LC2236]

- winlogon.exe 进程可能会在 twi3.dll 上意外关闭。

[#LC2244]

- 操作系统在 picadm.sys 上出现错误，并在蓝屏上显示错误检测代码 0x00000022。
[#LC2265]
- VDA 服务器在 icardd.dll 上遇到错误，显示蓝屏。
[#LC2277]
- 在已安装 Hotfix Rollup Pack 5 的服务器上，远程桌面服务在充当 XML Broker 的 XenApp 6.5 服务器上可能会意外退出。
[#LC2376]
- 服务器可能会在 ctxdvc.sys 上遇到致命异常，并在蓝屏上显示检测错误代码 0x0000003B。
[#LC2444]
- Windows 资源管理器进程可能会在已发布的桌面会话期间意外终止，并可能显示错误消息“Windows Explorer has stopped working”（Windows 资源管理器已停止运行）。
[#LC2733]
- 销售点设备通过 COM 端口重定向到共享桌面时，共享桌面上运行的第三方软件可能无法识别此设备。
[#LC2775]
- 服务器可能会在 icausb.sys 上遇到错误，并在蓝屏上显示消息 PNP_DETECTED_FATAL_ERROR。
[#LC2811]
- wfshell.exe 进程可能意外关闭并阻止建立新连接。
[#LC3024]
- 服务器可能会在 vdtw30.dll 和 win32k.sys 上遇到致命异常，并在蓝屏上显示错误。
[#LC3103]
- 启用“客户端剪贴板重定向”策略时，wfshell.exe 进程可能会意外关闭。
[#LC3249]
- VDA 可能会在 vdtw30.dll 上遇到致命异常，并显示蓝屏。
[#LC3354]
- csrss.exe 进程可能会在 vdtw30.dll 上遇到致命异常，并包含停止代码 0x7e。
[#LC3482]

用户体验

- 此修复解决了版本 4.200 之前的 Receiver for Windows 桌面组合重定向通过 LAN 连接使用时的可用性问题。
[#LC1507]
- 此版本包含的一项增强功能允许使用 Receiver for HTML5 或 Receiver for Chrome 登录的用户通过远程会话和本地设备传输文件。

[#LC3014]

用户界面

- 此修复解决了下列关于应用程序在 Receiver for Windows 工具栏上显示方式的问题：

- 单个应用程序可以显示默认的 Receiver 图标而非应用程序的特定图标。
- 当多个应用程序运行在同一会话中时，它们可以按组显示在单个默认 Receiver 图标下。

[#LC0094、#LC1181]

- 启用“远程控制组合框”策略的情况下，用户使用 iPad 在 Receiver for iOS 中打开 Internet Explorer 时，远程组合框的气球不能正常显示。例如，单击组合框 B 时，气球显示为来自组合框 A。

[#LC1800]

- 无缝应用程序窗口的某些边框可能隐藏。当“GetWindowRect”函数无法获取窗口的正确大小并因此导致显示的窗口坐标不包含边框时会出现此问题。

[#LC1943]

- 用户从 Receiver for Windows 或移动 Receiver 打开已发布的应用程序时，“自动显示键盘”可能不显示。

[#LC2747]

其他

- 此修复提供了与 Microsoft 更新的一般兼容性改进功能。

[#LA5638]

- 此修复解决了基础组件中的内存问题。

[#LA5765]

- 尝试检索许可证服务器地址时，许可证服务器名称中显示错误的字符。

[#LC0363]

- 此修复解决了 WMI 进程中的句柄泄漏问题。

[#LC0662]

- 此修复解决了特殊文件夹重定向和字符数超过 205 个的文件名存在的稳定性问题。

[#LC0719]

- 此修复解决了基础组件中的内存问题。

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

- 用户启动 Excel 的已发布版本，然后通过“使用电子邮件发送”来发送文件时，用户在 Outlook 中键入文本，然后数次按空格键后，插入中文字符而非空格。

[#LC1155]

- 用户使用 Receiver for Mac 登录时，带有 USB 适配器的 COM 端口签名板不能正常使用。用户使用 Receiver for Windows 登

录时，签名板不能正常使用。

[#LC1396]

- CTXCDF 中的句柄泄漏导致 Windows Management Instrumentation Provider Service (WMIPRVSE.exe) 停止，并在事件日志中显示 EventID 5612。

[#LC1764]

- 此修复解决了 Citrix Audio Redirection Service (CtxAudioSvc) 中的内存泄漏问题。

[#LC2054]

- Powershell 命令“Get-BrokerSession | Format-List -Property SessionState”显示不正确的会话状态。例如，断开的连接显示为活动会话。

[#LC2080]

- 本版本支持 Framehawk 虚拟通道，这是新 ICA 虚拟通道，扩展了 Citrix HDX 技术以改进使用经常出现数据包丢失和延迟问题的宽带无线连接时的用户体验。有关详细信息，请参阅[新增功能](#)。

[#LC2782]

- 此增强功能提供了事件 ID 1480 的消息，并包括以下信息：

- 已发布的应用程序名称
- 命令行
- 用户 ID
- 用户域
- 客户端名称
- 会话 ID

[#LC2830]

- 在 VDA 的 C: 驱动器的根目录中创建 sessionchange.log 文件后无法将其禁用或移动到其他位置。

[#LC3621]

已知问题

Nov 02, 2016

Feature Pack 3

- In recent Firefox versions, the browser add-ons for URL redirection are disabled by default, so you must enable them. This is a third-party issue. [#283844]
- Users cannot open some files from redirected folders such as Downloads and Desktop, if they downloaded the files using the Edge browser within a Windows 10 VDA. To work around this issue, use Internet Explorer to download files. [#591635]
- The following registry keys are not present after upgrading a brokerless VDA to FP3, and must be added manually [#591030]:
 - hklm\software\citrix\VirtualDesktopAgent\HaModeComputerName
 - hklm\software\citrix\VirtualDesktopAgent\HaModeTimeEnd
- When using Receiver for Windows with a Wacom pen device on a Windows 10 VDA, if you lose the network connection, the cursor may fail to display once the session is reconnected (using Automatic Client Reconnection, ACR). To work around this issue, disconnect and reconnect the session. [#593516]
- Persistent CDF tracing is not available if the Citrix Telemetry Service is running. To work around this issue, temporarily disable the Citrix Telemetry Service. [#593347]
- There is an error in XenApp and XenDesktop 7.6 FP3, when you update an existing Windows 10 master image on XenServer to use the XenServer hotfix [XS65ESP1010](#) with XenTools. [#593985]
- Apps which are published from a Windows 10 VDA do not display their full borders when maximized [#593596]. To work around this issue, registry keys can be set for each application class. For example, to work around the behavior for Paint, you create the following key:

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

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

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

Note that you must add the Class Name for an application, not the application name. For example, for Paint add **MSPaintApp**. You can get the Class Name using a tool such as Spy++. For more information on Seamless Configuration Settings, see [CTX101644](#).

- Some "Learn more" help links may not redirect correctly to the current Product Documentation site (docs.citrix.com). To work around this issue, access the Product Documentation site directly. All documentation for 7.6 FP3 and earlier 7.6 feature packs is on the new Product Documentation site under [XenApp and XenDesktop 7.6](#).
- Temp files from the standalone VDA installation package remain in the VDA Operating System's temp folder after completion of the install. To work around this issue, manually delete the temporary files. [#594829]

Feature Pack 2

Rich-text formatting is lost when text is copied from an application within the session and pasted into an application outside of the session, and when it is copied from outside the session and pasted inside. [#0538497]

Host names (the host part of the fully qualified domain name, or FQDN) longer than 15 characters are truncated, preventing successful VDA registration. [#0544120]

Linux VDA registration status fails to change on the endpoint when resuming from hibernation. During a power cycle event, the endpoint fails to obtain an IP address, which prevents it from registering. [#0569303]

The Linux VDA does not currently support DNS lookup of realm mappings. [#0557555]

KDE-based applications appear unavailable in Linux VDA sessions. [#0568006]

Linux VDAs running RedHat Enterprise fail to connect if Quest authentication service is enabled and SELinux is in enforcement mode. [#0551761]

Citrix Receiver for Windows (Version 4.3) and Citrix Receiver for iOS Classic (Version 6.0) are the only Receiver platforms currently supported.

Session Recording/Smart Auditor are not supported with Framehawk.

This release of Framehawk is designed for standard XenApp and XenDesktop workloads, such as knowledge worker and office apps. It has not been tested with high-end graphic intensive environments

Multi-monitor clients are not supported at this time.

The Vd3dn.dll may crash on the client under test conditions, such as modifying bandwidth and latency on a WAN emulator in-session. Disconnect the session before making any changes to WAN emulator settings, then reconnect; Framehawk recalibrates during session handshake.

If the Framehawk policy and the legacy policy are both enabled, the Framehawk policy takes precedence on Workstation operating systems (for example, Windows 7); for RDS environments (for example, Windows 2012 R2), the legacy policy is applied.

For Windows 7 environments, the initial XenDesktop 7.6 FP2 release of Framehawk does not provide touch support (for example, pinch and zoom), despite this functionality being present in a Windows 7 OS. Additionally, Windows 2008 R2 does not provide touch support.

For the StoreFront 3.0 known issues and fixed issues, see the StoreFront 3.0 documentation [known issues](#) and [fixed issue](#).

For the HDX RealTime Optimization Pack 1.8 known issues and fixed issues, see the Optimization Pack 1.8 documentation [known and fixed issues](#).

Feature Pack 1

- When Machine Creation Services (MCS) or Provisioning Services creates a VDA with configured master image and Microsoft Message Queuing (MSMQ) installed, the VDA has the same QMId as the MSMQ. This might cause various issues, such as:
 - Sessions might not be recorded even if the recording agreement is accepted.
 - The session logoff signal might not be received by the Session Recording server, which leads to the session always in Live status. [#528678]
- The workaround to create a unique and persistent QMId for each VDA is to use a script. To use the script, do the following:

1. Make sure the execution policy is set to RemoteSigned or Unrestricted, in PowerShell.
Set-ExecutionPolicy RemoteSigned
2. Create a scheduled task and set the trigger as At system startup and run with SYSTEM account on the Provisioning Services or MCS master image machine.
3. Add the command as a startup task.
powershell.exe -file C:\GenQMID.ps1

警告

Editing the registry incorrectly can cause serious problems that may require you to reinstall your operating system. Citrix cannot guarantee that problems resulting from the incorrect use of Registry Editor can be solved. Use Registry Editor at your own risk. Be sure to back up the registry before you edit it.

Summary of the script:

1. Generate the QMId based on the hash value of the machine FQDN.
2. Stop related services, including CitrixSmAudAgent and MSMQ.
3. Set the QMId in the registry.
4. Start services that stopped previously to apply QMId's change.

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

Try {
    # Get UUID of machine

    $strUUID = (Get-WmiObject -Class Win32_ComputerSystemProduct | Select-Object -Property UUID).UUID

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

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

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

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

```
# Calculate MD5 hash of FQDN

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

# Set md5 digest as QMID

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

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

}

# Write new_QMID into registry

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

# Restart MSMQ to adopt new QMID

# Get dependent services

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

Restart-Service -force MSMQ

# Start dependent services

if ($depServices -ne $null) {

    foreach ($depService in $depServices) {

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

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

            Start-Service $depService.Name

        }

    }

}
```

- When recording a session with a resolution higher than or equal to 4096 x 4096, there might be fragments in the recording appearance. [#524973]
- When you change your XenApp or XenDesktop license type, the change does not take effect immediately for Session Recording. Workaround: Restart the VDA machine. [#532393]
- You might receive an Installation failed error in the following two cases. You can ignore the message, but to avoid receiving the message, restart the machine before reinstalling the Session Recording components. [#544579]
 - Uninstalled the Session Recording components, and then reinstalled them without restarting the machine.
 - Installation failed and rollback happened, and then you tried to reinstall the Session Recording components without restarting the machine.
- Limitation for Session Recording to support the Pre-Launched application sessions [BUG0561109]
 - Problem:
 - If the active policy tries to match the application name, the application launched in the pre-launched session will not be matched, which results in the session not being recorded.
 - If the active policy records every application, when the user logs into the Windows Receiver (at the same time the pre-launched session is established) a notification for recording will appear and the empty session and any applications that will be launched in this session later will be recorded.
 - Workaround:
 - Publish the applications in separate Delivery Groups according to their recording policy. Do not use the application name as the recording condition. This will ensure pre-launch sessions will be recorded. However, notifications will still appear.
- When recording a session with a resolution higher than or equal to 4096 x 4096, there might be fragments in the recording appearance. [#524973]
- When you change your XenApp or XenDesktop license type, the change does not take effect immediately for Session Recording. Workaround: Restart the VDA machine. [#532393]
- You might receive an Installation failed error in the following two cases. You can ignore the message, but to avoid receiving the message, restart the machine before reinstalling the Session Recording components.
 - Uninstalled the Session Recording components, and then reinstalled them without restarting the machine.
 - Installation failed and rollback happened, and then you tried to reinstall the Session Recording components without restarting the machine.
- When recording a session with a resolution higher than or equal to 4096 x 4096, there might be fragments in the recording appearance. [#524973]
- When you change your XenApp or XenDesktop license type, the change does not take effect immediately for Session Recording. Workaround: Restart the VDA machine. [#532393]

- You might receive an Installation failed error in the following two cases. You can ignore the message, but to avoid receiving the message, restart the machine before reinstalling the Session Recording components, and then reinstalled them without restarting the machine.
- Installation failed and rollback happened, and then you tried to reinstall the Session Recording components without restarting the machine.
- When recording a session with a resolution higher than or equal to 4096 x 4096, there might be fragments in the recording appearance. [#524973]
- When you change your XenApp or XenDesktop license type, the change does not take effect immediately for Session Recording. Workaround: Restart the VDA machine. [#532393]
- You might receive an Installation failed error in the following two cases. You can ignore the message, but to avoid receiving the message, restart the machine before reinstalling the Session Recording components, and then reinstalled them without restarting the machine.
- Installation failed and rollback happened, and then you tried to reinstall the Session Recording components without restarting the machine.

For the Citrix Licensing 11.12.1 known issues and fixed issues, see the Citrix Licensing 11.12.1 documentation [known and fixed issues](#).

For the HDX RealTime Optimization Pack 1.7 known issues and fixed issues, see the Optimization Pack 1.7 documentation [known and fixed issues](#).

XenApp 7.6 and XenDesktop 7.6

The following note applies to any workaround that suggests changing a registry entry:

Caution: Editing the registry incorrectly can cause serious problems that may require you to reinstall your operating system. Citrix cannot guarantee that problems resulting from the incorrect use of Registry Editor can be solved. Use Registry Editor at your own risk. Be sure to back up the registry before you edit it.

- On Server OS machines, the CreateAnonymousUsersApp tool may not delete all anonymous user accounts from the local machine; all anonymous users are local users, not Active Directory users. Because the tool deletes passwords and profiles, any anonymous-user account names that are not deleted are no longer useable. To delete unwanted anonymous-user account names locally on the server, use Manage User Accounts or Computer Manager. [#4999679]
- To configure a nonstandard HTTP/SOAP port for the Universal Print Server web service, use PowerShell cmdlets to change the session printer policy. For information about configuring Group Policy settings, see the Group Policy SDK usage section in the *About the SDK* document. To set the policy value, use:
Set-ItemProperty LocalGpo:\Computer\Unfiltered\Settings\ICA\Printing\UniversalPrintServer\UpsHttpPort -name Value -Value <portnumber>. [#268593]

- After a Hyper-V host is unpaused, Microsoft System Center Virtual Machine Manager (VMM) might not update the overall host state immediately. This can affect the use of Machine Creation Services (MCS). If one Hyper-V host reports a paused state, that host will not be used to provision Virtual Machines (VMs); if all hosts report a paused state, catalog creation will fail. As a workaround, manually refresh the parent cluster node or Hyper-V host node. Also, running environment tests on the host will identify hosts reporting a paused state. [#285696]
- Brokering hosted applications on Desktop OS machines is not supported using the Remote Desktop Protocol (RDP). [#377108]
- This release does not support mounting an .iso file when Client Drive Mapping (CDM) is configured for Windows 8 sessions. [#333111]
- If the Citrix Universal Print Server fails because of bad drivers, try running those drivers under printer driver isolation. For information about how to configure printer driver isolation, see 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]
- The Enhanced Desktop Experience policy setting does not affect pre-existing user or administrator profiles. As a workaround, delete all pre-existing profiles before enabling/disabling the setting, and delete the profile used for VDA installation (after installing the VDA). If the built-in administrator's account was used for (Virtual Delivery Agent (VDA) installation, you cannot delete the profile. In that case, the user can choose the Citrix Enhanced Desktop Theme when logged on. [#363736]
- If monitor resolution is affected because the VDA desktop session resolution exceeds the client monitor resolution, change the resolution setting on the VDA desktop to the highest supported value for the attached monitor. Alternatively, you can detach the monitor cables from the VDA graphics card. [#365877]
- Screen savers and the power-save option are disabled in sessions. Edit the registry and create the following DWORD value:
HKLM\Software\Citrix\Graphics\SetDisplayRequiredMode = 0
This change does not prevent the remote machine screen saver or power save mode from coming on. If the power save mode comes on, the remote session is not updated until the user provides input (mouse/keyboard), but the screen will not be blanked. [#380550]
- Attempting to use a PowerShell SDK New-ProvScheme command or any MCS command from a remote machine before setting the host admin address might result in an error. Set the admin address using the Set-HypAdminConnection command before running the New-ProvScheme command. [#336902]
- This release does not support using this product with Microsoft RemoteFX vGPU feature in a Hyper-V host. Instead, use RDP to access RemoteFX functionality such as Hyper-V vGPU. [#375577]
- Director reports the number of GPU machines that have failed to start in the Machine Failure panel on the Dashboard page. However, this information is not displayed when you drill down to view details on the Filters page.

You can also view this information in the Historical trends graph. [#0434722]

- If more GPU machines are provisioned or assigned than the XenServer resource pool can support of the GPU type specified, GPU machines may fail to start. Users assigned to a machine (including those using Personal vDisk) that failed to start may not be able to access their virtual desktop through StoreFront. [#0434505]
- If you power on a GPU machine and it fails to start, you may see the following error message:

```
Exception Failure in PowerOn, PGPU_INSUFFICIENT_CAPACITY_FOR_VGPU
```

This indicates that there are insufficient GPU Resources to start another machine that uses the GPU. [#434509]

- If multiple users log on to a Desktop OS machine using RDP and ICA protocols and a user locks his/her RDP session, users connecting through ICA cannot log on to the session. To prevent this issue, users should disconnect or log off RDP sessions when they are finished. [#392311]
- When you publish an application with a Windows 8 VDA host and the application shows non-ASCII characters in the notification area, other unrelated characters might also appear in the notification area after reconnection. To resolve this issue, log off from the session and then launch the application again. [#387963]
- The ability to disable session sharing through a registry entry is not supported in this release. As a result, session sharing is always enabled. If the registry key is set to disable session sharing, the first application launches, but subsequent applications do not launch. There is no workaround for this issue. [#383718]
- When a user reconnects to a disconnected session on Windows Server 2012, there is a memory leak in the Desktop Window Manager. If users connect and disconnect frequently to long-lived sessions, such as in the case of an employee using the same session from work and home, this can cause memory resources on the server to be reduced, eventually leading to slow response times and possibly even server failure. This is a third-party issue in the Microsoft code. For information, see: <http://support.microsoft.com/kb/2855336> (to be published in July 2013). [#374261]
- The Help About topics fail when working through Windows PowerShell 3.0. This is a third-party issue with Microsoft. Other help topics are unaffected. [#408866]
- When upgrading from XenDesktop 5 to this release (or to XenDesktop 7.1 or XenDesktop 7.0), a hosted application assigned to both Shared and Private Delivery groups produces a Delivery Group

incompatibility error during upgrade. To avoid this error, do not assign hosted applications to both Shared and Private Delivery Groups. [#419424]

- If the Profile management feature is enabled, logon scripts for sessions running Windows Server 2012 R2 or Windows 8.1 are delayed by five minutes by default. Once the session is available, the Logon duration for logon scripts step is not available in Director. The delay is controlled by the Configure Logon Script Delay policy (Enabled = 0). [#407978]
 - Desktops may not launch on computers running Windows 8.1 with Microsoft Software Update Management installed. To avoid this, in the Windows 8.1 **Taskbar and Navigation properties** dialog box, on the **Navigation** tab, select all the options in the **Start screen** section. [#408439]
 - If you receive the error "You cannot access this session because no licenses are available" when you try to connect through Remote Desktop Protocol to a brokered session, disable these settings in C:\inetpub\wwwroot\Citrix\Store\App_Data\default.ica: [#422212 and #403855]
 - RDPConnection=false
 - RDP-RedirectDrives=false
 - RDP-RedirectDynamicDrives=false
 - The value for the Persistent Cache Threshold policy setting is labelled incorrectly, as Kbps, in Studio. The correct value is bits per second (bps). [# 429478]
 - When selecting a custom date range for a Configuration Logging report, the calendar displays might be incorrect. [#452399]
 - Machines provisioned on Amazon Web Services or CloudPlatform will not be suspended if their Delivery Group Power Management setting is set to Suspend when disconnected. [#453780]
 - Connection times (maximum connection timer, connection timer, and disconnect timer) might fail to work on Windows Server 2012 machines containing Windows Server OS VDAs, causing unexpected session time-out behavior. [#471698]
 - When the Auto-create generic universal printer policy setting is enabled, the Citrix Print Manager Service (CpSvc.exe) does not create the Citrix Universal Printer generic printer. Hotfixes that address the issue are available as Knowledge Center articles CTX141565 and CTX141566.
 - User connections to VDAs running Windows Server 2012 R2 might fail if the **Maximum allowed color depth** Citrix policy setting is enabled. The **Maximum allowed color depth** does not apply to VDA that use a Windows Display Driver Model (WDDM) driver as the primary display driver, such as VDAs running Windows Server 2012 R2.
-
- Error 1904. Module C:\Program Files (x86)\Citrix\System32\rpm.dll failed to register. HRESULT -2147010895. Contact your support personnel occurs when installing or upgrading the Virtual Delivery Agent (VDA) using the Citrix MetaInstaller, the installation of the third-party Microsoft Visual C++ 2005 Runtime component fails, rolls back, and then falsely notifies the MetaInstaller that the installation completed successfully. Because of this, the MetaInstaller continues to install any remaining VDA components until it reaches the post-install component initialization stage. You cannot continue without the Microsoft Visual C++ 2005 Runtime component being present at this stage. To work around this issue, rerun the MetaInstaller VDA installation, which then successfully reruns the Microsoft Visual C++ 2005 Runtime component installation. Alternatively, see <http://support.microsoft.com/?kbid=947821> and follow Microsoft's documented solution. [#489633]
 - Installing a Virtual Delivery Agent for Server OS might fail with error 1935 because of backward compatibility errors in the Microsoft Visual C++ 2005 Redistributable. See the Microsoft website or run Windows Update to check for fixes. [354833]
 - After a successful VDA for Windows Server OS installation, but before the machine restarts, the Windows event log might contain several error messages (such as TermService 1035 or 1036, indicating the Terminal Server listener stack was down or the session creation failed). If there are no other installation failure indicators, you can safely ignore those event log messages. [#374134]
 - When upgrading from XenDesktop 5.x, make sure that the XenDesktop 5.x Desktop Studio is closed before running the upgrade. Otherwise, Studio may close unexpectedly during the upgrade. [#389374]
 - When upgrading, an administrator who was disabled in XenDesktop 5.6 might move to the later version without a role or scope. Check the Administrators display in Studio and edit administrators, as needed. [#394765]
 - If you enable the Profile management feature and users find that their default Windows 8 applications (such as Weather, News, and Bing) start the first time they log on, but not after subsequent logons, you might have to reconfigure this feature. This issue has been observed in environments where the user profile is not persisted and if the folder AppData\Local has not been excluded (the default). As a workaround, add the folders AppData\Local\Packages and AppData\Local\Microsoft\Application Shortcuts as exclusions. [#394802]
 - During VDA installation or upgrade on Windows 7, you might see a Windows dialog box prompting you to Restart the computer to apply changes. Click Restart Later to continue the upgrade; do not select Restart Now. [#396553]
 - Installing VDAs through Active Directory Group Policy using individual MSIs is not recommended and might fail. Citrix recommends using the startup scripts provided on the product installation media, as described in [Install or remove Virtual Delivery Agents using scripts](#). [#383432, #372136]
 - The optimization phase of the Virtual Delivery Agent (VDA) installation might take a long time to complete. In some tests, it has taken about half an hour and may take longer. These instances occur when installing the VDA on an image running a Windows operating system, if, on the installation wizard Features page, Optimize Performance has been selected. This causes the Microsoft Native Image Generator (Ngen) to run. If this occurs, allow the installation process to finish. Citrix recommends that you run Ngen on your VDA base image prior to provisioning virtual desktops to avoid delays caused when it runs in the background of provisioned virtual desktops. [#381437]
 - During upgrade, if an error message mentioning PICAlsPorticaV2 entry point not found appears during an upgrade, it can be safely ignored. Complete the upgrade process and restart the machine when prompted. [#423947]
 - The VDA for Windows Desktop OS might not install on evaluation versions of Windows 8. The Installation Options screen displays the message "Cannot be installed on this operating system." The issue occurs because the installation program is incorrectly identifying Windows 8 evaluation versions as unsupported operating systems. A hotfix that addresses the issue is available as Knowledge Center article CTX139660. The hotfix lets you patch the installer before running it on Windows evaluation versions.
 - During product installation, machines created by Provisioning Services might fail if .NET Framework 3.5 is not present prior to the installation. To work around this problem, make sure that all NET Framework versions 2.0, 3.0, 3.5, 4.0, 4.5, and 4.5.1 are installed. [#442639, #447851]
-
- XenDesktop 7.6 includes a new version of the volume worker package for CloudPlatform. Citrix highly recommends updating the volume worker template in your deployment with this new version. When updating, please note that in-place upgrades for this package are not supported. Therefore, you must first fully uninstall the previous package or build a new volume worker template from scratch. [493211]
-
- After upgrading from XenDesktop 5.6 FP1, the Remote PC Access Service administrator name may not display correctly. This does not affect operations. [#437948]
 - When an office machine has been instructed to hibernate, a subsequent Remote PC Access session launch may fail. As a workaround for desktop machines that display an error message, try launching the session again. For laptops that display a persistent grey reconnecting screen, restart the PC (remotely from the administrator console using Force Shutdown/Force Restart, or locally with the power button); this can result in data loss. [#441154]
 - When relying on the Wake-up Proxy rather than Intel Active Management Technology (AMT) or Wake on LAN packets, a machine might fail to wake up. This is a Microsoft System Center Configuration Manager issue. [#441412]
 - Starting a VDA from a user device with a smart card reader might fail if the user previously started the VDA from that device and selected Disconnect from the Desktop Viewer. In this case, the user might see the smart card credential screen or the message 'Reading smart card.' As a workaround, choose one of the following:
 - Remove and reinsert the smart card in the reader
 - Click Cancel. Then, in the VDA, press Ctrl+Alt+Del. [#322301]

- Attempting to launch both StoreFront and Studio causes the Citrix console to exit unexpectedly after XenApp and XenDesktop software is installed on a single Windows 2008 R2 SP1 machine. This occurs when launching both StoreFront and Studio; at the end of the installation, from shortcut menus, or if you open StoreFront in the console first, and then open Studio. To work around this issue, force the Native Image Generator (Ngen) to update the .Net native images. To do this, open a command prompt, and then navigate to c:\windows\microsoft.net\framework64\<v2.0.50727>. Run ngen update /force. This may take several minutes.
Note: The framework64 version number may vary slightly, but it should always start with 2.0.
[#490819]
- Director does not correctly report licensing errors, and Studio is unable to communicate with the license server. This issue may occur if the license server address for a Delivery Site changes while multiple instances of Studio are open and managing the same XenDesktop or XenApp site. To workaround this issue, refresh licensing data in the site by closing all open instances of Studio, reopening a single instance of Studio, and then navigating to the Licensing node. [#492971]
- When using Machine Creation Services (MCS) to provision machines on VMware vSphere, the combination of CPU sockets and cores on the provisioned machines reflects the combination of CPU sockets and cores on the base image used to create the machine catalogs. During MCS catalog creation, if the number of Virtual CPUs selected is higher than the maximum possible on the host, then provisioned machines are created with the maximum possible sockets and cores for that host, without indication to the user. [#331269]
- When using System Center Virtual Machine Manager in a pure IPv6 environment, and using Machine Creation Services to create machine catalogs, all VMs have both IPv4 and IPv6, even if the master VM is configured without IPv4 in the TCP/IP stack. This is a third party issue with Microsoft, and there is no workaround. [#371712]
- When using VMware vSphere in an IPv4 and IPv6 environment with VMware ESX hypervisors configured with VMXNET3 network adapters, all VMs have both IPv4 and IPv6, even if the master VM is configured without IPv4 in the TCP/IP stack. This is a third-party issue with VMware, and there is no workaround. [#371712]
- Long machine catalog names and long storage path names might cause a disk attach error in the VMM job window. Microsoft has identified a maximum limit of 255 characters on the length of the file path for VM resources. This issue has been seen when using local storage on a standalone Hyper-V host, due to the long file path used to store the VMs; however, it is not limited standalone Hyper-V hosts. [#359673]
As a workaround:
 - Create VMM MCS catalogs with short names, especially when the storage is accessed using a long path.
 - Shorten the path to the storage used to store the VMs.
- To change a database to one that was previously used, you must use the SDK; it cannot be done from Studio. [#355993]
To switch to the Configuration Logging database:
 - Set-LogDbConnection -DataStore 'Logging' -DBConnection \$null
 - Set-LogDbConnection -DataStore 'Logging' -DBConnection '<new database connection string>'
 To switch to the monitoring secondary database:
 - Set-MonitorDbConnection -DataStore 'Monitor' -DBConnection \$null
 - Set-MonitorDbConnection -DataStore 'Monitor' -DBConnection '<new database connection string>'
 For example, the new database connection string could be:


```
'Server=dbserver;Initial Catalog = dbname; Integrated Security = True'
```
- When you use MCS to provision machines on your hypervisor platform, CPUs are added but no cores. If you change the CPU value during catalog creation, the products licensed on a per CPU basis might need more licenses. For example, your master image has one CPU and four cores and you change the CPU value during catalog creation. If, during MCS catalog creation, you select more CPUs than the maximum possible on the host, provisioned machines have the maximum number supported for that host and you are not notified of the difference. For example, a Desktop OS machine can use only two physical CPUs; thus, you will see only two even if more are assigned.
As a workaround, ensure your master image VM has the same virtual CPU configuration that you want to deploy for the catalog. [#331274]
- When Lync 2013 client is delivered from a Desktop OS machine or Server OS machine, the video chat feature does not work. See XenDesktop 7.x, XenApp 6.x and Citrix Receiver 4.x Support for Microsoft Lync 2013 VDI Plug-in, (<http://support.citrix.com/article/CTX138408>) for information about using Lync 2013. [#371818]
- When launching a seamless application on a Windows server, the window might not have the Aero theme, even when Enhanced Desktop Experience is enabled.
 - Users launching only seamless applications never get the Aero theme.
 - Users with a mix of seamless applications and desktops get the Aero theme after establishing the first desktop session; then, seamless applications have the Aero theme.
 As a workaround, set the Citrix Enhanced Desktop theme in the default user profile; all users on that VDA get the Enhanced Desktop Experience for their seamless applications. The theme is part of the VDA install and must be set for all VDAs. [#348812]
- Studio messages sent to a Windows 8 machine will display on the Windows 8 Desktop, and not the Windows default (formerly called Metro) display mode. The user must switch to Desktop mode to see the message. This is a third party issue and there is no workaround. [#387356]
- If a Site contains more than one hosting infrastructure object with the same name, the Studio display might not be correct. When you create hosting infrastructure objects (for example, networks and storage), it is best practice to specify a unique name for each. [#384959]
- After using Delegated Administration to create an administrator with a new scope, refresh the Studio display. Otherwise, you might receive a permissions error when you log on as the new administrator and attempt to create a new connection or resource. [#386634]
- Applications hosted on Desktop OS machines with random assignments might fail to open after the loading dialog box disappears. This issue occurs when the time the application takes to start exceeds the default one-minute time-out and the session exits automatically. [#389025]
As Administrator, change the base image and re-provision with a changed timeout value as follows:

Locate the registry key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\wfshell\TWI (Create the key if it is not available.)

Name: ApplicationLaunchWaitTimeoutMS

Type: REG_DWORD

Data: Required additional time-out, in milliseconds

Note: Specifying a value of less than 10000 reverts to 10000 because 10 seconds is the minimum override.
- To discover and set up applications in Delivery Groups, make sure that there are some machines that are registered and can be powered on. For an App-V application to be discovered, you need to configure App-V Publishing in Studio. [#393676]
- After a user launches the first App-V application, launching a second App-V application too quickly might fail. If this occurs, the user should wait a few minutes so that the initial synchronization can complete, and then launch the second application again. For details about this issue, see [CTX138056](#). [#397521]
- If the antivirus program BitDefender is installed on a VDA, you might not be able to create machine catalogs. There is no workaround for this issue. [#392705]
- After upgrading to XenDesktop 7.5, you cannot create a connection to CloudPlatform or Amazon Web Services (AWS) if the site is configured with a connection to an on-premise hypervisor. You must create a new 7.5 Site. For a hybrid deployment (which includes cloud and non-cloud Sites), you must create separate Sites that share the same StoreFront site. [#454114]
- Creating a machine catalog fails in an environment running VMM 2012 SP1 and Hyper-V 2012 when using a ODX-enabled Storage Area Network. Follow the instructions in <http://support.citrix.com/article/CTX139333> to resolve this issue. [#424040]

- The Citrix policy setting **ICA listener connection timeout** applies only to Virtual Delivery Agent 5.0, 5.5, and 5.6 Feature Pack 1, even though the Studio policy editing window says this setting also applies to 7.0 Desktop OS, 7.1 Desktop OS, 7.5 Desktop OS, and 7.6 Desktop OS.
 - In Studio, the description of the Citrix policy setting **Launching of non-published programs during client connection** incorrectly states that this setting specified whether to launch initial applications or published applications through ICA or RDP on the server. This setting specifies only whether to launch initial applications or published applications through RDP on the server.
 - Citrix policies might not be applied correctly when filtered by tags. To prevent this, apply the hotfix at [CTX142439](#). [#529165]
-
- The User-based application usage data in the Hosted Application Usage reports, is not accurate for time ranges that exceed the last seven days. This occurs because application data beyond the last seven days is deleted. To avoid using incorrect User-based application usage data, only use data reported within the last seven days. [504642]
 - Director displays the Unexpected error. Check your network connection or view server event log for further information popup error on the Trends and Filters page because the XenDesktop database name includes spaces. The API calls used by Director do not support connecting to a database that has a name containing spaces. Follow database naming requirements when renaming a database. [494339]
 - After upgrading, Director does not persist an administrator's UI.GlobalSearchResults settings. To work around this issue, manually edit the UI.GlobalSearchResults setting in the web.config file after upgrading. [484066]
 - Hotfix inventory may take several hours to update when large numbers of hotfixes are applied at the same time to environments with more than 20,000 desktops. [489604]
 - Historical information is preserved after you have deleted and then re-created a Delivery Group from the same Session Machine Catalog. The data then incorrectly displays that historical information on the Trends graph and table. To create Delivery Groups without preserving historical trend data, do not re-create the Delivery Group until after you have deleted the machine catalog and created the new catalog. [#480010]
 - When navigating from the Filters page to the User Device page, user names may not display. If Delivery Controllers cannot access a domain containing user accounts because of Active Directory restrictions, those full user names may not appear on the User Device page. To work around this issue, ensure that the domain where XenDesktop is installed is trusted by the domain to which the end user belongs. [#479517]
 - If all Sites within an existing Site Group do not have assigned delegated administrators, the search for machines and end-point devices within the Site Group fails. To allow an administrator to search across all Sites within a Site Group, add the administrator to the Delegated Administrator group that has permissions to the Sites. [#491740]
 - When monitoring Windows XP virtual desktops running WinRM 2.0 for users with VDAs earlier than 7, you must change the WinRM port listening order. For details about how to change the setting to 5985.80, see [Advanced configuration](#). [#273609]
 - You cannot edit an existing Site Group using the Director Configuration Tool. To work around this issue, open the .xml configuration file from within the Director Configuration folder, and then modify the Site Group setting. [#491681]
 - Using Active Directory Users and Computers to configure users with logon scripts fails and data does not appear in the Logon Duration panel. Instead, to configure users with logon scripts, you must use a Group Policy. [#393259]
 - The logon duration data from a first-time logon to a Personal vDisk VDA might not be collected or displayed in Director. For subsequent logons, data appears normally. [#383941]
 - When you run a console session from a Windows Server 2012 desktop, navigating to the user details page in Director might result in an error and no data will be displayed.
- As a workaround, register the VDA, log off the current session, and log in to the VDA again. [#388513]
- In the Infrastructure panel of the Dashboard page, Director displays "Not Available" for Citrix CloudPlatform-based host connections, Amazon Web Services, Hyper-V, and Microsoft System Center Configuration Manager and does not provide status information. [#449806, #446397]
 - Logon duration does not update in Director's Dashboard view if users launch a hosted application. [#386860]

Note: For the latest updates to HDX Flash compatibility, refer to [CTX136588](#).

- Configuring HDX MediaStream redirection on a new machine fails. To work around this issue, reboot the end-point machine after installing Receiver. [#494741]
- Sessions fail if using a certificate with the SHA512 algorithm for an SSL/TLS connection. This occurs when using Receiver for HTML5 to launch sessions with Server OS machines running Windows Server 2012 R2. To avoid this issue, do not use the SHA512 algorithm for an SSL/TLS certificate in this type of configuration. [#487284]
- With GPU pass-through and NVIDIA Kepler cards, the first connection attempt may fail for a HDX 3D Pro user device with three or four monitors. If this happens, attempt to connect again. [#422049]
- When a user attempts to connect, change the screen resolution, or change the size of the session screen, the screen may flicker or display as black. This may occur if larger resolutions are selected and the video driver cannot allocate enough memory at boot time to support the resolution. You can view the current resolution setting in the session window by selecting Advanced Settings from the Screen Resolution dialog. To allow the driver to allocate enough memory to support higher resolutions, increase the amount of memory for the VDA. Use the following as guidelines: Windows 7: Sum-of-all-monitors (width x height x 4 BytesPerPixel x 2 BackBuffers) Win8: Sum-of-all-monitors (width x height x 4 BytesPerPixel x 3 BackBuffers). [#494671]
- If the USB Redirection policy is enabled, USB storage and audio devices do not work when the user chooses to redirect their devices. This only occurs on Server OS machines running Windows 2012 and later. To exclude audio and storage devices from the list of devices, use the Client USB Device Redirection Rule policy. For additional information on USB redirection rules and configurations, see <http://support.citrix.com/article/CTX137939>. [#479578]
- When the user attempts to launch a session from a Server OS machine, Receiver displays the "The connection to <app name/desktop name> failed with status (1030)" error message on the user's device after the Session Reliability Connections policy is changed from disabled to enabled by the administrator. To work around this issue, the administrator must restart the VDA for the policy to take effect. [#486073]
- When using Remote PC to run a Lync 2013 session remotely, attached web cameras on client devices are not listed in Lync 2013 in the user's session. To work around this issue, the user can use Device Manager to disable the web cameras on their remote machines. [#482807]
- Even though webcams might support H.264 compression, this release does not support hardware compression, so you must use software compression for those webcams. To do this, add a registry entry on user devices at HKEY_CURRENT_USER\Software\Citrix\HdxRealTime; add a DWORD registry name DeepCompress_ForceSWEncode. When set to 1, software compression is used. By default, this setting is off and hardware compression is used. [#357356]
- HDX RealTime Webcam video lags if the video resolution is higher than 720p (1280x720). [#350187]
- When using HDX Flash Redirection continuously, the session might become unresponsive. [#350085, 361926]
- HDX RealTime Webcam supports most raw formats supported by a webcam, but in rare cases, if the webcam has an unsupported format, that webcam might not work as the Citrix HDX Webcam. [#338318]
- Multiple duplicate images might be seen intermittently when using Citrix HDX Webcam with some models of webcams. [#367322]
- HDX RealTime Webcams are not supported for these applications:
 - Citrix GoToMeeting when hosted on Server OS Machines with Windows 2012 operating systems. [#346430]
 - GoToMeeting (on any platform) if the webcam is attached after a meeting has started. [#346140]
 - Microsoft Lync 2013 and Adobe Connect with VDAs on Windows 8, Windows 8.1, and Windows 2012 operating systems. [#340784, 348506, 459732]
 - Microsoft Office Communications Server (OCS) video calls if the Webcam is attached after the call is in progress. [#370236]
 - Microsoft Silverlight. This is an intermittent issue. As a workaround, on the user device, enable the legacy codec by adding a DWORD registry key value name EnableDeepcompress_Client at HKEY_CURRENT_USER\Software\Citrix\HdxRealTime and setting it to 0. [#379779]
 - 64-bit video conferencing applications. Video compression for 64-bit applications is not supported. [#366515]

- When using HDX 3D Pro with the XenDesktop 5.6 Feature Pack 1 Virtual Desktop Agent on Windows XP virtual desktops, during the first connection, the Fine Drawing (2D) check box is selected and sometimes greyed out. This is due to delayed registry updates, which cause the Config tool UI to pick up incorrect default values during initialization. [#353031]
As a workaround, disconnect and reconnect the session.

If the problem persists, clear previous session information by deleting settings under the following registry entry:

HKey_Current_Users\Software\Citrix\HDX3D\BitmapRemotingConfig

- On Windows XP, Citrix HDX webcam might not be detected. As a workaround, install Microsoft Visual C++ 2005 Service Pack 1 Redistributable Package from <http://www.microsoft.com/en-us/download/details.aspx?id=14431> website and try again. [#382733]
- When viewing the Display Adapters node from the Device Manager console applet, the Standard VGA Graphics Adapter appears in the list with a yellow exclamation point (yellow bang). You can ignore this warning because it does not affect functionality. This warning occurs because a legacy model XPDM display driver (Standard VGA Graphics Adapter) is not allowed to load when a new model WDDM display driver (Citrix Display Driver) is installed. [#339390]
- Users might experience issues when attempting to play media files on Windows 8 user devices. This is because this product fails to register the correct default program for client-side content fetching protocols used to stream media files to user devices. As a workaround for Microsoft Media Streaming (MMS) and Real Time Streaming (RTS) protocols, change the default program used for playing media files from Windows Media Player to Citrix CSF Handler. There is no workaround for the Hypertext Transfer Protocol (HTTP). [#328805]
- TWAIN redirection fails on hosted shared desktops and applications. This is a third party issue related to TWAIN applications that require TWAIN binaries to be located in certain paths. [#300854, 340999]
As a workaround, on your Windows Server 2012 machine running the VDA, copy these files to the following locations:
 - Copy "twain_32.dll" to the "\WINDOWS" directory of the User profile (for example, copy twain_32.dll into the folder: "%USERPROFILE%\Windows\").
 - Copy "twain_32.dllmui" into the "\WINDOWS\en-US\" directory of the User profile (for example, copy twain_32.dllmui into the folder: "%USERPROFILE%\Windows\en-US\").
- The 64-bit Windows Media Player or QuickTime player cannot play some video files using server-side rendering when HDX MediaStream Windows Media Redirection is disabled. As a workaround, use the 32-bit version of Windows Media Player. [#384759]
- Universal Print Server printers selected in the virtual desktop do not appear in the Devices and Printers window in Windows Control Panel. However, when users are working in applications, they can print using those printers. This issue occurs only on Windows Server 2012, Windows 10 and Windows 8 platforms. [#335153]
- With GPU pass-through and NVIDIA Kepler cards, the first connection attempt may fail for a HDX 3D Pro user device with three or four monitors. The second connection should be successful. [#422049]
- The user's Windows computer stops responding when a GoToMeeting session using a webcam configured for USB redirection is started in a Remote PC Access session with an Intel Core i7 processor-based computer. If this occurs, restart the user's computer and restart the Remote PC Access session. The session resumes where the disconnection occurred. To avoid this occurrence, use HDX webcam video compression instead of USB redirection. For details, see <http://support.citrix.com/proddocs/topic/xendesktop-7/hd-new-graphics-video.html>. [#423284]
- User devices running Receiver for HTML5 might be unable to connect to a Server OS machine running Windows Server 2012 R2. To avoid this issue:
 - Use an existing machine, rather than a machine created with Machine Creation Services (MCS) or Provisioning Services, as the Windows Server OS machine.
 - If you plan to use machines created with MCS, on the master image for the catalog, edit registry and create the following DWORD value:
Caution: Editing the registry incorrectly can cause serious problems that may require you to reinstall your operating system. Citrix cannot guarantee that problems resulting from the incorrect use of Registry Editor can be solved. Use Registry Editor at your own risk. Be sure to back up the registry before you edit it.
HKEY_LOCAL_MACHINE\Software\Citrix\GroupPolicy\Defaults\lcaPolicies\AcceptWebSocketsConnections = 1
HKEY_LOCAL_MACHINE\Software\Citrix\GroupPolicy\Defaults\lcaPolicies\AllowDesktopLaunchForNonAdmins = 1
HKEY_LOCAL_MACHINE\Software\Citrix\GroupPolicy\Defaults\lcaPolicies\WebSocketsPort = 8008
Because the WebSockets port number is set by editing the registry, it is not necessary to enable the WebSockets connections Citrix policy for the catalog.
- If you are using a machine created with Provisioning Services, follow the recommendations in [CTX139265](#). [#424064]
- When HDX Flash Redirection is used with a dual-monitor setup and the Flash content window in full-screen on one monitor, clicking anywhere else on the screen might cause the Flash content window to lose focus and be hidden behind the session window. This is behavior as designed for HDX Flash Redirection. If this happens, users can make the Flash content window visible again by changing the session window from full-screen mode to window mode using the Citrix Desktop Viewer toolbar. [#567132]
- Errors may occur when installing the Delivery Controller if the most recent version of the license server is not installed. When upgrading to version 7.6, install the latest version of the License Server before installing any other core components. [#510425]
- When licensing is uninstalled and then reinstalled and a read-only product administrator attempts to view Licensing information in Studio, Studio displays the error "You do not have permissions to perform this operation." A read-only administrator does not have the permission to trust a new License Server. As a workaround, a full license administrator must go to the Licensing node and authenticate the License Server. [#380982]
- If you installed the License Server without successfully configuring it (with the post installation License Server Configuration tool), any subsequent License Server upgrade fails. As a workaround, ensure that every License Server installation is configured with the post-installation License Server Configuration tool. [#377079]
- When you try to start the License Administration Console or the Simple License Service, a blank page might display if the Internet Explorer Enhanced Security Configuration is enabled and the License Administration Console or the Simple License Service is not in the Trusted Sites zone. Workaround: Disable Internet Explorer Enhanced Security Configuration. [#382429]
- If port 8083 is in use when you install or upgrade the product, the License Server configuration and installation fail with a License Server Configuration Failed error. As a workaround, check the event log to ensure the error is actually "Port in use." If it is:
 - Uninstall the License Server by double-clicking on the CTX_Licensing.msi in the x64\Licensing folder on the product installation media.
 - Run the installer again. It displays some components as Partially installed. Click Install and the installer completes the installation and configures any necessary product components.
 - Manually install the License Server and specify port numbers that do not conflict with other applications on the machine. [#390815]
- The user list for the License Administration Console and the Citrix Simple License Service Web page does not support non-ASCII characters in user/group names. Due to this limitation, on a Russian operating system, the BUILTIN Administrators group is not added to the user list because it is created with non-ASCII characters. This issue applies to both fresh installs and upgrades. Any users belonging to the BUILTIN Administrators group in an earlier release of XenDesktop and the Simple License Service will not have access to the License Administration Console or the Simple License Service after an upgrade.
As a workaround, add ASCII-character versions of Russian users/groups names post-installation using the License Administration Console interface. Alternatively, install the license server on one of the other supported operating systems. [#395305]
- When you install or upgrade the product and have Perpetual (permanent) licenses, Studio might display your licenses with an expiration date of 01/01/2000. You can ignore this expiration date and launch your desktops and applications. [#402975]
- In a Windows 8 or Windows Server 2012 hosting environment, if Local App Access is enabled and the extension for a client hosted app does not have a File Type Association (FTA), FTA redirection fails. The user is prompted to select "Look for an app in the Store" or "More options." [#372834]
As a workaround, use one of these methods on the VDA master image:

- Rename the DelegateExecute registry value for HKEY_CLASSES_ROOT\Unknown\shell\openas\command\DelegateExecute.
 - Use Notepad to open a file with the extension. FTA redirection will work for subsequent attempts.
 - URL redirection is disabled, by default, by Microsoft on Windows Server 2012. To enable it, disable Internet Explorer enhanced configuration mode. [#356260]
 - Changes to Local App Access properties during a session do not take effect automatically. As a workaround, log off and log back on. [#357488]
 - If Local App Access applications are launched on Windows 8 or Windows 2012 platforms, those VDAs cannot be launched from the Modern shell. As a workaround, close the local application and then launch the VDA application. [#359670]
 - Shellhook.dll is not loaded with Receiver for Windows 3.4 and earlier when local applications are launched. As a workaround, change the value of the registry key LocalAppInit_DLLs to 1 under HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows. If the workaround is not used, client to server FTA redirection does not work correctly. [#356130]
 - URL redirection might fail for Internet sites that have a pop-up blocker enabled. As a workaround, disable the pop-up blocker. However, for security reasons, disabling pop-up blocker is not recommended. [#371220]
 - When Aero mode is enabled for a VDA session, there are inconsistencies between the VDA local applications and the client-hosted applications (for example, ALT+Tab; flashing taskbar entries, jump list, live preview). For compatibility, disable the Aero mode. [#361043]
 - Flash Redirection has compatibility issues such as a WMP black screen and flash pseudo-container window out of bounds. As a workaround, disable Flash redirection. [#360182]
 - The Desktop Composition Redirection graphics feature is disabled when Local App Access is enabled. To use that feature, disable Local App Access. [#377386]
 - With Session Reliability and Local App Access enabled, if network connectivity is disrupted after you click Home in the product toolbar and display the client's desktop, the VDA session might not display the last screen shown before the network disruption. Instead, only the product toolbar and the client's desktop appear. On the toolbar, only the Disconnect option works. [#357769]
 - When Local App Access is enabled and a user changes the product session to full screen before or while playing a media file, some or all of the image does not appear. To work around this issue, relaunch the session. [#402702]
-
- After installing Citrix Desktop Lock on a domain-joined Windows 8 machine and restarting it, the desktop background may become black and a My Computer window may be displayed instead of the usual Start screen. To log off the machine, type logoff in the address bar of the window, and press Enter. For more information about configuring Windows 8 machines with Desktop Lock, see [Desktop Lock](#). [#329075]
 - If a Windows 8 or Windows 7 machine fails and the user attempts to restart it by pressing Ctrl+Alt+Del, the Windows Security dialog on the local desktop opens. This issue occurs even when the Citrix Desktop Lock software is installed and configured. To resolve this issue, select Start Task Manager to display the Restart dialog. [#337507]
 - The Citrix Desktop Lock does not redirect Adobe Flash content to domain-joined user devices. The content can be viewed but is rendered on the server, not locally. As a workaround, configure Adobe Flash redirection for server-side content fetching to pass the content from the server to the user device. This issue does not occur on non-domain-joined devices or when the content is viewed with the Desktop Viewer. [#263092]
 - Failure to start the virtual desktop agent (VDA) on the user device may occur if the device is running Windows 7 with Desktop Lock and Receiver for Windows Enterprise 3.4. This may occur if the user is or is not connected to the Internet. The error message, "No Internet Connectivity," may appear when this occurs. This is a third party issue with Microsoft. For a potential resolution, see <http://technet.microsoft.com/en-us/library/cc766017>. [#408642]
-
- Handle leaks by wfica32.exe can occur on the user device when playing Windows Media Player files continuously in a Windows 32-bit XP client session. [#378146]
 - Disconnect button is not available to users. As a workaround, provide a shortcut to the TSDicon.exe utility, which is included with the operating system; this will allow them to disconnect from sessions. [#362937]
 - The user may continue to have access to the remote PC after attempting to disconnect. This occurs when the user selects the Disconnect command from the Start menu. There is a delay after Disconnect is selected. If the user presses Ctrl+Alt+Del during this delay, the VDA on the remote PC remains available for several minutes before disconnecting. During that time, the VDA on the user device freezes until the remote VDA disconnects. [#322301]
 - For VDAs installed on Windows 7 and Windows 8 platforms, two mouse pointers might be visible: one movable and one locked to the UI. This is a third party issue with the NVidia driver. As a workaround, you can disable NVidia GRID technology (formerly known as VGX) by running MontereyEnable.exe -disable -reset, and then restarting the machine. [#307921]
 - The Microsoft Desktop Composition feature has a scalability cost for VDAs. For users requiring maximum scalability, Desktop Composition should be disabled by Microsoft policy for any user not using Desktop Composition Redirection. [#386602]
 - In certain scenarios, when users attempt to unlock a locked session locally, the session relocks itself repeatedly. This issue might occur if the user unplugs or powers off a keyboard locally connected to a Remote PC Access machine during a remote ICA session. As a workaround, users should relaunch the session remotely, disconnect the session, and then log on again from the console. [#382554]
 - For VDAs earlier than 7, users' data might not appear in Director even after you correctly configure Windows Remote Management (WinRM) for these VDAs. If this issue occurs, restart the WinRM service and the data should display in Director as expected. [#392047]
 - In secure environments, a new VDA might fail to register with a Controller. Specifically, when installing a VDA containing a local security policy setting that allows only administrators to access a computer from the network, the VDA installs but cannot register with a Controller. Instead, a warning is issued that user access rights are not properly configured. For more information, see [CTX117248](#). [#336203]
 - Receiver for Windows users cannot log on to stores using pass-through authentication, even though the domain pass-through authentication method is enabled in the StoreFront authentication service. To resolve this issue, run the command Set-BrokerSite -TrustRequestsSentToTheXmlServicePort \$true from a Windows PowerShell command prompt on the Controller. [#330775]
 - ICA Roundtrip checks are not supported when legacy graphics mode has been specified using a policy for Windows Server 2012 VDAs. There is no workaround for this issue. Note that Windows 8 VDAs do not support legacy mode, so they are not affected by this issue. [#394824]
 - With Windows Server 2012 R2 and older Domain Controllers, if you require users to change their password on next login, an error occurs. To work around this problem, remove the Microsoft update KB2883201. [#438725]
 - When users connect directly to VDAs using RDP, the Controller sometimes mistakenly reports additional sessions that do not exist, and these non-existent sessions remain in a Connected state in Studio until the VDA is restarted. [#385823]
 - When Passthrough Authentication is implemented for Citrix Storefront on FireFox and Chrome browsers, users are prompted for credentials when launching applications. [#441487]
-
- For Citrix Receiver for Web with a domain pass-through configuration, when a user logs on to a device using Smartcard and then launches a published desktop, the connection may fail. To work around this issue, edit the default ICA file in the store (for example, in C:\inetpub\wwwroot\Citrix\StoreApp_Data\default.ica), and add the following line to the [Application] section:
DisableCtrlAltDel=False
[#452813]

To view personal vDisk related known issues, see [Personal vDisks](#).

To view Provisioning Services issues, see [About Provisioning Services 7.0, 7.1, and 7.6](#)

本版本中不提供的功能

May 04, 2016

当前未提供或不再支持以下功能。

- **启动触控优化桌面** - 此设置已针对 Windows 10 计算机禁用。有关详细信息，请参阅 [移动体验策略设置](#)。
- **低于 128 位的安全 ICA 加密** - 在 7.x 之前的版本中，可以使用安全 ICA 加密客户端连接，以实现基本加密、40 位、56 位和 128 位加密。在 7.x 版本中，安全 ICA 加密仅适用于 128 位加密。
- **旧版打印** - 7.x 版本不支持以下打印功能：
 - 向后兼容 DOS 客户端和 16 位打印机，包括旧版客户端打印机名称。
 - 支持连接到 Windows 95 和 Windows NT 操作系统的打印机，包括增强型扩展打印机属性和 Win32FavorRetainedSetting。
 - 启用或禁用自动保留和自动恢复的打印机的功能。
 - DefaultPrnFlag。这是服务器上用于启用或禁用自动保留和自动恢复的打印机的一项注册表设置，存储在服务器上的用户配置文件中。
- **Secure Gateway** - 在 7.x 之前的版本中，Secure Gateway 是用于在服务器和用户设备之间提供安全连接的选项。NetScaler Gateway 是用于确保外部连接安全的替代选项。
- **重影用户** - 在 7.x 之前的版本中，管理员通过设置策略控制用户对用户重影操作。在 7.x 版本中，重影最终用户是 Director 组件的一项集成功能，该功能使用 Microsoft 远程协助来允许管理员重影和解决与已交付的无缝应用程序和虚拟桌面有关的问题。
- **电源和容量管理** - 在 7.x 之前的版本中，可以使用电源和容量管理功能来降低电源消耗和管理服务器容量。Microsoft Configuration Manager 工具替代了此功能。
- **Flash v1 重定向** - 不支持第二代 Flash 重定向的客户端（包括 3.0 之前的 Receiver for Windows 版本、11.100 之前的 Receiver for Linux 版本以及 Citrix Online Plug-in 12.1）将回退到服务器端呈现，以实现旧版 Flash 重定向功能。7.x 版本中包括的 VDA 支持第二代 Flash 重定向功能。
- **本地文本回显** - 此功能与早期的 Windows 应用程序技术结合使用，用于在高延迟连接中，在用户设备上加速显示输入文本。由于图形子系统和 HDX SuperCodec 的功能得以增强，因此 7.x 版本中不提供此功能。
- **Smart Auditor** - 在 7.x 之前的版本中，Smart Auditor 允许记录用户会话的屏幕活动。7.x 版本中不提供此组件。在 7.6 Feature Pack 1 中，被 Session Recording 取代。
- **单点登录** - 此功能可以保证密码安全，但在 Windows 8 和 Windows Server 2012 环境中不受支持。Windows 2008 R2 和 Windows 7 环境中仍支持此功能，但 7.x 版本不包括此功能。可以在 Citrix 下载 Web 站点找到此功能：<http://citrix.com/downloads>。
- **Oracle 数据库支持** - 7.x 版本需要使用 SQLServer 数据库。
- **运行状况监视与恢复 (HMR)** - 在 7.x 之前的版本中，HMR 可以在服务器场中的服务器上运行测试，以监视它们的状态并发现任何运行状况风险。在 7.x 版本中，Director 从 Director 控制台监视整个基础结构并提供警报，从而提供了一种从中央位置查看系统运行状况的方式。
- **自定义 ICA 文件** - 自定义 ICA 文件用于从用户设备（通过 ICA 文件）直接连接到特定计算机。在 7.x 版本中，此功能默认处于禁用状态。但在正常情况下，可以通过本地组将其启用。在 Controller 不可用时，还可以在高可用性模式中使用该功能。
- **Management Pack for System Center Operations Manager (SCOM) 2007** - 该管理包之前使用 SCOM 监视场的活动，但不支持 7.x 版本。
- **CNAME 功能** - 在 7.x 之前的版本中，CNAME 功能默认处于启用状态。如果部署依赖于 CNAME 记录进行 FQDN 重新路由并且使用 NETBIOS 名称，则可能会失败。在 7.x 版本中，Delivery Controller 自动更新是其替代功能，该功能可以动态更新 Controller 的列表，并且还可以在向站点添加 Controller 或从站点删除 Controller 时自动向 VDA 发送通知。Controller 自动更新功能在 Citrix 策略中默认处于启用状态，但可以通过创建策略禁用该功能。
或者，也可以在注册表中重新启用 CNAME 功能，以继续使用现有部署并允许 FQDN 重新路由和使用 NETBIOS 名称。有关详细信息，请参阅 [CTX137960](#)。

- **快速部署向导** - 在 7.x 之前的 Studio 版本中，利用此选项可以对完整安装的 XenDesktop 部署进行快速部署。7.x 版本中提供简化的全新安装和配置工作流程，不需要再使用“快速部署”向导选项。
- **用于实现自动管理的 Remote PC Service 配置文件和 PowerShell 脚本** - Remote PC 现在已集成到 Studio 和 Controller 中。
- **Workflow Studio** - 在 7.x 之前的版本中，Workflow Studio 是用于 XenDesktop 的工作流组合的图形界面。7.x 版本不支持此功能。
- **颜色深度** - 在 7.6 之前的 Studio 版本中，此选项位于交付组“用户设置”页面，用于设置交付组的颜色深度。在 7.6 版本中，可以使用 New-BrokerDesktopGroup 或 Set-BrokerDesktopGroup PowerShell cmdlet 设置颜色深度。
- **在客户端连接期间启动非发布程序** - 在 7.x 之前的版本中，此 Citrix 策略设置指定是否在服务器上通过 ICA 或 RDP 启动初始应用程序或已发布的应用程序。在 7.x 版本中，此设置仅指定是否在服务器上通过 RDP 启动初始应用程序或已发布的应用程序。
- **桌面启动** - 在 7.x 之前的版本中，Citrix 策略设置指定非管理员用户是否可以连接到桌面会话。在 7.x 版本中，非管理员用户必须属于 VDA 计算机的直接访问用户组才能连接到此 VDA 上的会话。**桌面启用**设置使 VDA 直接访问用户组的非管理员用户可以使用 ICA 连接连接到 VDA。**桌面启动**设置不影响 RDP 连接；不管是否启用此设置，VDA 直接访问用户组的用户都可以使用 RDP 连接与 VDA 建立连接。
- **COM 端口映射** — COM 端口映射可允许或阻止访问用户设备上的 COM 端口。在之前版本中，COM 端口映射默认处于启用状态。在 XenDesktop 和 XenApp 的 7.x 版本中，COM 端口映射默认处于禁用状态。有关详细信息，请参阅[使用注册表配置 COM 端口和 LPT 端口重定向设置](#)。
- **LPT 端口映射** — LPT 端口映射控制旧版应用程序对 LPT 端口的访问。在之前版本中，LPT 端口映射默认处于启用状态。在 7.x 版本中，LPT 端口映射默认处于禁用状态。
- **PCM 音频编解码器** — 在 7.x 版本中，只有 HTML5 客户端支持 PCM 音频编解码器。
- **支持 Microsoft ActiveSync。**
- **针对旧版本的代理支持** — 包括：
 - Microsoft Internet Security and Acceleration (ISA) 2006 (Windows Server 2003)。
 - Oracle iPlanet Proxy Server 4.0.14 (Windows Server 2003)。
 - Squid Proxy Server 3.1.14 (Ubuntu Linux Server 11.10)。

系统要求

Nov 02, 2016

In this article:

[Session Recording - XenApp 7.6 FP1, FP2, and LTSR](#)

[Delivery Controller](#)

[Database](#)

[Studio](#)

[Director](#)

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

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

[Hosts / virtualization resources](#)

[Active Directory functional level support](#)

[HDX - Desktop Composition Redirection](#)

[HDX - Windows Media delivery](#)

[HDX - Flash Redirection](#)

[HDX 3D Pro](#)

[HDX - Video conferencing requirements for webcam video compression](#)

[HDX - Other](#)

[Universal Print Server requirements](#)

[Other requirements](#)

The system requirements in this document were valid when this product version released. System requirements components not covered here (such as StoreFront, host systems, receivers and plug-ins, and Provisioning Services) are described in their respective documentation.

Important: Review [Prepare to install](#) before beginning an installation.

Unless otherwise noted, the component installer deploys software prerequisites automatically (such as .NET and C++ packages) if they are not detected on the machine. The Citrix installation media also contains some of this prerequisite software.

The installation media contains several third-party components. Before using the Citrix software, check for security updates from the third party, and install them.

The disk space values are estimates only, and are in addition to space needed for the product image, operating system, and other software.

If you install all the core components (Controller with SQL Server Express, Studio, Director, StoreFront, and Licensing) on a single server, you need a minimum of 3 GB of RAM to evaluate the product; more is recommended when running an environment for users. Performance will vary depending on your exact configuration, including the number of users, applications, desktops, and other factors.

Important: After you install XenApp on a Windows Server 2012 R2 system, use the Kerberos Enable Tool (XASsonKerb.exe) to ensure the correct operation of Citrix Kerberos authentication. The tool is located Support > Tools > XASsonKerb folder on the installation media; you must have local administrator privileges to use the tool. To ensure correct Kerberos operation, run xassonkerb.exe -install from a command prompt on the server. If you later apply an update that changes the registry location HKLM\System\CurrentControlSet\Control\LSA\OSConfig, run the command again. To see all available tool options, run the command with the -help parameter.

Session Recording - XenApp 7.6 FP1, FP2, and LTSR

Session Recording is available in English, French, German, Japanese, Spanish, and simplified Chinese. All Session Recording components that connect to each other must be the same language edition; mixed-language installations are not supported.

The English-language edition of Session Recording is supported on English, Russian, traditional Chinese, and Korean operating systems. The French, German, Japanese, Spanish, and simplified Chinese editions of Session Recording are supported on operating systems in their respective languages.

Session Recording Administration components

You can install the Session Recording Administration components (Session Recording Database, Session Recording Server, and Session Recording Policy Console) on a single server or on separate servers.

Session Recording Database

Supported operating systems:

- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012
- Microsoft Windows Server 2008 R2 with Service Pack 1

Requirements:

- Microsoft SQL Server 2008 R2 SP3 Shared Management Objects and Microsoft SQL Server 2008 R2 SP3 System CLR Types packages. You can download the packages at <http://www.microsoft.com/en-us/download/details.aspx?id=44272>.
- Microsoft SQL Server 2014 (Enterprise and Express editions), Microsoft SQL Server 2012 (Enterprise and Express editions) with Service Pack 2, or Microsoft SQL Server 2008 R2 (Enterprise and Express editions) with Service Pack 3
- .NET Framework Version 3.5 Service Pack 1
- .NET Framework Version 3.5 Service Pack 1

Session Recording Server

Supported operating systems:

- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012Microsoft Windows Server 2008 R2 with Service Pack 1

Requirements:

Before starting the Session Recording installation, you must install some prerequisites. Open the Server Manager and add the IIS role. Select the following options:

- Application Development -
ASP.NET 3.5 (other components are automatically selected. Click Add to accept required roles)
- Security - Windows Authentication
- Management Tools - IIS 6 Management Compatibility
 - IIS 6 Metabase Compatibility
 - IIS 6 WMI Compatibility
 - IIS 6 Scripting Tools
 - IIS 6 Management Console
- NET Framework Version 3.5 Service Pack 1
- If the Session Recording Server uses HTTPS as its communications protocol, add a valid certificate. Session Recording uses HTTPS by default, which Citrix recommends.
- Microsoft Message Queuing (MSMQ), with Active Directory integration disabled, and MSMQ HTTP support enabled.

Session Recording Policy Console

Supported operating systems:

- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012
- Microsoft Windows Server 2008 R2 with Service Pack 1

Requirements:

- .NET Framework Version 3.5 Service Pack 1

Session Recording Agent

Install the Session Recording Agent on every XenApp and XenDesktop server on which you want to record sessions.

Supported operating systems:

- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012
- Microsoft Windows Server 2008 R2 with Service Pack 1

Requirements:

- Microsoft Message Queuing (MSMQ), with Active Directory integration disabled, and MSMQ HTTP support enabled
- .NET Framework Version 3.5 Service Pack 1

Session Recording Player

Supported operating systems:

- Microsoft Windows 8.1
- Microsoft Windows 8
- Microsoft Windows 7 with Service Pack 1

For optimal results, install Session Recording Player on a workstation with:

- Screen resolution of 1024 x 768
- Color depth of at least 32-bit
- Memory: 1GB RAM (minimum). Additional RAM and CPU/GPU resources can improve performance when playing graphics intensive recordings; especially when there are a lot of animations in the recordings.

The seek response time depends on the size of the recording and your machine's hardware specification.

Requirements:

- .NET Framework Version 3.5 Service Pack 1

Session Recording Administration components

You can install the Session Recording Administration components (Session Recording Database, Session Recording Server, and Session Recording Policy Console) on a single server or on separate servers.

Session Recording Database

Supported operating systems:

- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012
- Microsoft Windows Server 2008 R2 with Service Pack 1

Requirements:

- .NET Framework Version 3.5 Service Pack 1 (Windows Server 2008 R2 only) or .NET Framework Version 4.5.1 or 4.6.

Session Recording Server

Supported operating systems:

- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012Microsoft Windows Server 2008 R2 with Service Pack 1

Requirements:

Before starting the Session Recording installation, you must install some prerequisites. Open the Server Manager and add the IIS role. Select the following options:

- Application Development:
 - ASP.NET 4.5 on Server 2012 and Server 2012 R2, ASP.NET on Server 2008 R2 (other components are automatically selected. Click Add to accept required roles)
- Security - Windows Authentication
- Management Tools - IIS 6 Management Compatibility
 - IIS 6 Metabase Compatibility
 - IIS 6 WMI Compatibility
 - IIS 6 Scripting Tools
 - IIS 6 Management Console
- NET Framework Version 3.5 Service Pack 1 (Windows Server 2008 R2 only) or .NET Framework Version 4.5.1 or 4.6.
- If the Session Recording Server uses HTTPS as its communications protocol, add a valid certificate. Session Recording uses HTTPS by default, which Citrix recommends.
- Microsoft Message Queuing (MSMQ), with Active Directory integration disabled, and MSMQ HTTP support enabled.

Session Recording Policy Console

Supported operating systems:

- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012
- Microsoft Windows Server 2008 R2 with Service Pack 1

Requirements:

- .NET Framework Version 3.5 Service Pack 1 (Windows Server 2008 R2 only) or .NET Framework Version 4.5.1 or 4.6.

Session Recording Agent

Install the Session Recording Agent on every XenApp and XenDesktop server on which you want to record sessions.

Supported Windows operating systems:

- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012
- Microsoft Windows Server 2008 R2 with Service Pack 1

Requirements:

- Microsoft Message Queuing (MSMQ), with Active Directory integration disabled, and MSMQ HTTP support enabled
- .NET Framework Version 3.5 Service Pack 1 (Windows Server 2008 R2 only) or .NET Framework Version 4.5.1 or 4.6.

Session Recording Player

Supported operating systems:

- Microsoft Windows 8.1
- Microsoft Windows 8
- Microsoft Windows 7 with Service Pack 1

For optimal results, install Session Recording Player on a workstation with:

- Screen resolution of 1024 x 768
- Color depth of at least 32-bit
- Memory: 1GB RAM (minimum). Additional RAM and CPU/GPU resources can improve performance when playing graphics intensive recordings; especially when there are a lot of animations in the recordings.

The seek response time depends on the size of the recording and your machine's hardware specification.

Requirements:

- .NET Framework Version 3.5 Service Pack 1 or .NET Framework Version 4.5.1 or 4.6.

Supported operating systems:

- Windows Server 2012 R2, Standard and Datacenter Editions
- Windows Server 2012, Standard and Datacenter Editions
- Windows Server 2008 R2 SP1, Standard, Enterprise, and Datacenter Editions

Requirements:

- Disk space: 100 MB. Connection leasing (which is enabled by default) adds to this requirement; sizing depends on the number of users, applications, and mode (RDS or VDI). For example, 100,000 RDS users with 100 recently-used applications require approximately 3 GB of space for connection leases; deployments with more applications may require more space. For dedicated VDI desktops, 40,000 desktops require at least 400-500 MB. In any instance, providing several GBs of additional space is suggested.
- Microsoft .NET Framework 3.5.1 (Windows Server 2008 R2 only).
- Microsoft .NET Framework 4.5.1 (4.5.2 and 4.6 are also supported).
- Microsoft .NET Framework 4.6.1
- Windows PowerShell 2.0 (included with Windows Server 2008 R2) or 3.0 (included with Windows Server 2012 R2 and Windows Server 2012).
- Visual C++ 2005, 2008 SP1, and 2010 Redistributable packages.

Supported Microsoft SQL Server versions for the Site Configuration Database (which initially includes the Configuration Logging Database and the Monitoring Database):

- SQL Server 2016, Express, Standard, and Enterprise Editions.
- SQL Server 2014 through SP2, Express, Standard, and Enterprise Editions.
- SQL Server 2012 through SP3, Express, Standard, and Enterprise Editions. By default, SQL Server 2012 SP1 Express is installed when installing the Controller, if an existing supported SQL Server installation is not detected.
- SQL Server 2008 R2 SP2 and SP3, Express, Standard, Enterprise, and Datacenter Editions.

The following database features are supported (except for SQL Server Express, which supports only standalone mode):

- SQL Server Clustered Instances
- SQL Server Mirroring
- SQL Server 2012 AlwaysOn Availability Groups

Windows authentication is required for connections between the Controller and the SQL Server database.

For information about the latest supported database versions, see [CTX114501](#).

Supported operating systems:

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

Requirements:

- Disk space: 75 MB
- Microsoft .NET Framework 4.6.1
- Microsoft .NET Framework 4.5.1 (4.5.2 and 4.6 are also supported)
- Microsoft .NET Framework 3.5 SP1 (Windows Server 2008 R2 and Windows 7 only)
- Microsoft Management Console 3.0 (included with all supported operating systems)
- Windows PowerShell 2.0 (included with Windows 7 and Windows Server 2008 R2) or 3.0 (included with Windows 8.1, Windows 8, Windows Server 2012 R2, and Windows Server 2012)

Supported operating systems:

- Windows Server 2012 R2, Standard and Datacenter Editions
- Windows Server 2012, Standard and Datacenter Editions
- Windows Server 2008 R2 SP1, Standard, Enterprise, and Datacenter Editions

Requirements:

- Disk space: 50 MB.
- Microsoft .NET Framework 4.5.1 (4.5.2 and 4.6 are also supported).
- Microsoft .NET Framework 3.5 SP1 (Windows Server 2008 R2 only)
- Microsoft Internet Information Services (IIS) 7.0 and ASP.NET 2.0. Ensure that the IIS server role has the Static Content role service installed. If these are not already installed, you are prompted for the Windows Server installation media, then they are installed for you.
- Supported browsers for viewing Director:
 - Internet Explorer 11 and 10.
Compatibility mode is not supported for Internet Explorer. You must use the recommended browser settings to access Director. When you install Internet Explorer, accept the default to use the recommended security and compatibility settings. If you already installed the browser and chose not to use the recommended settings, go to Tools > Internet Options > Advanced > Reset and follow the instructions.
 - Firefox ESR (Extended Support Release).
 - Chrome.

Supported operating systems:

- Windows 10, Enterprise Edition. This applies to XenApp and XenDesktop 7.6 FP3 VDA Standalone installations using the Standard VDA only. The following are not supported on Windows 10 in 7.6 FP3:
 - HDX 3D Pro
 - GPU acceleration
 - Desktop Composition Redirection
 - Legacy graphics mode

- Secure Boot
- Publishing Universal Windows apps using VM Hosted Apps
- Citrix Profile Management
- Windows 8.1, Professional and Enterprise Editions
- Windows 8, Professional and Enterprise Editions
- Windows 7 SP1, Professional, Enterprise, and Ultimate Editions

To use the Server VDI feature, you can use the command line interface to install a VDA for Windows Desktop OS on a supported server operating system; see [Server VDI](#) for guidance.

- Windows Server 2012 R2, Standard and Datacenter Editions
- Windows Server 2012, Standard and Datacenter Editions
- Windows Server 2008 R2 SP1, Standard, Enterprise, and Datacenter Editions

Requirements:

- Microsoft .NET Framework 4.6.1
- Microsoft .NET Framework 4.5.1 (4.5.2 and 4.6 are also supported)
- Microsoft .NET Framework 3.5.1 (Windows 7 only)
- Microsoft Visual C++ 2005, 2008, and 2010 Runtimes (32-bit and 64-bit). This applies to XenApp and XenDesktop 7.6, 7.6 FP1 and FP2.
- Microsoft Visual C++ 2008, 2010 and 2013 Runtimes (32-bit and 64-bit). This applies to XenApp and XenDesktop 7.6 FP3 VDA Standalone installations.

Remote PC Access uses this VDA, which you install on physical office PCs.

Several multimedia acceleration features (such as HDX MediaStream Windows Media Redirection) require that Microsoft Media Foundation be installed on the machine on which you install the VDA. If the machine does not have Media Foundation installed, the multimedia acceleration features will not be installed and will not work. Do not remove Media Foundation from the machine after installing the Citrix software; otherwise, users will not be able to log on to the machine. On most Windows 8.1, Windows 8, and Windows 7 editions, Media Foundation support is already installed and cannot be removed. However, N editions do not include certain media-related technologies; you can obtain that software from Microsoft or a third party.

During VDA installation, you can choose to install the HDX 3D Pro version of the VDA for Windows Desktop OS. That version is particularly suited for use with DirectX and OpenGL-driven applications and with rich media such as video.

Supported operating systems:

- Windows Server 2012 R2, Standard and Datacenter Editions
- Windows Server 2012, Standard and Datacenter Editions
- Windows Server 2008 R2 SP1, Standard, Enterprise, and Datacenter Editions

The installer automatically deploys the following requirements, which are also available on the Citrix installation media in the Support folders:

- Microsoft .NET Framework 4.6.1
- Microsoft .NET Framework 4.5.1 (4.5.2 and 4.6 are also supported)
- Microsoft .NET Framework 3.5.1 (Windows Server 2008 R2 only)
- Microsoft Visual C++ 2005, 2008, and 2010 Runtimes (32-bit and 64-bit). This applies to XenApp and XenDesktop 7.6, 7.6 FP1 and FP2.

- Microsoft Visual C++ 2008, 2010 and 2013 Runtimes (32-bit and 64-bit). This applies to XenApp and XenDesktop 7.6 FP3 VDA Standalone installations.

The installer automatically installs and enables Remote Desktop Services role services, if they are not already installed and enabled.

Several multimedia acceleration features (such as HDX MediaStream Windows Media Redirection) require that the Microsoft Media Foundation be installed on the machine on which you install the VDA. If the machine does not have Media Foundation installed, the multimedia acceleration features will not be installed and will not work. Do not remove Media Foundation from the machine after installing the Citrix software; otherwise, users will not be able to log on to the machine. On most Windows Server 2012 R2, Windows Server 2012, and Windows Server 2008 R2 editions, the Media Foundation feature is installed through the Server Manager (for Windows Server 2012 R2 and Windows Server 2012: ServerMediaFoundation; for Windows Server 2008 R2: DesktopExperience). However, N editions do not include certain media-related technologies; you can obtain that software from Microsoft or a third party.

Supported platforms:

- XenServer.
 - XenServer 6.5 SP1
 - XenServer 6.5
 - XenServer 6.2 SP1 plus hotfixes (you must apply SP1 to enable application of future hotfixes)
 - XenServer 6.1
- VMware vSphere - No support is provided for vSphere vCenter Linked Mode operation.
 - VMware vSphere 6.0 Update 1
 - VMware vSphere 6.0
 - VMware vSphere 5.5 Update 3
 - VMware vSphere 5.5 Update 2
 - VMware vSphere 5.5 Update 1
 - VMware vSphere 5.5
 - VMware vSphere 5.1 Update 3
 - VMware vSphere 5.1 Update 2
 - VMware vSphere 5.0 Update 3
 - VMware vSphere 5.0 Update 2
 - VMware vCenter 5.5 / 6 appliance
- System Center Virtual Machine Manager - Includes any version of Hyper-V that can register with the supported System Center Virtual Machine Manager versions.
 - System Center Virtual Machine Manager 2012 R2
 - System Center Virtual Machine Manager 2012 SP1
 - System Center Virtual Machine Manager 2012
- Nutanix Acropolis 4.5 - Several XenApp and XenDesktop features are not available when using this platform; see [CTX202032](#) for details. For more information on the use of the product with Acropolis, see <https://portal.nutanix.com/#/page/docs>.

You can also deploy this product in the following cloud environments:

- Amazon Web Services (AWS)
 - You can provision applications and desktops on supported Windows server operating systems.
 - SQL Server 2012 Enterprise is not available on AWS.

- AWS does not offer desktop operating system instances.
- The Amazon Relational Database Service (RDS) is not supported.
- See the AWS documentation and [Deploy XenApp and XenDesktop 7.5 and 7.6 with Amazon VPC](#) for additional information.
- Citrix CloudPlatform
 - The minimum supported version is 4.2.1 with hotfixes 4.2.1-4.
 - Deployments were tested using XenServer 6.2 (with Service Pack 1 and hotfix XS62ESP1003) and vSphere 5.1 hypervisors.
 - CloudPlatform does not support Hyper-V hypervisors.
 - CloudPlatform 4.3.0.1 supports VMware vSphere 5.5.
 - See the CloudPlatform documentation (including the Release Notes for your CloudPlatform version) and [XenApp and XenDesktop concepts and deployment on CloudPlatform](#) for additional support and Linux-based system requirements information.

See [CTX131239](#) for updated hypervisor support information.

The following virtualization resource and storage technology combinations are supported for Machine Creation Services and runtime Active Directory account injection into VMs. Combinations marked with an asterisk (*) are recommended.

Virtualization resource	Local Disks	NFS	Block Storage	Storage Link
XenServer	Yes	Yes *	Yes	No
VMware	Yes (no vMotion or dynamic placement)	Yes *	Yes	No
Hyper-V	Yes	No	Yes * (requires Cluster Shared Volumes)	No

The Remote PC Access Wake on LAN feature requires Microsoft System Center Configuration Manager 2012. See [Configuration Manager and Remote PC Access Wake on LAN](#) for details.

The following functional levels for the Active Directory forest and domain are supported:

- Windows 2000 native (not supported for domain controllers)
- Windows Server 2003
- Windows Server 2008
- Windows Server 2008 R2
- Windows Server 2012
- Windows Server 2012 R2

The Windows user device or thin client must support or contain:

- DirectX 9
- Pixel Shader 2.0 (supported in hardware)
- 32 bits per pixel

- 1.5 GHz 32-bit or 64-bit processor
- 1 GB RAM
- 128 MB video memory on the graphic card or an integrated graphics processor

HDX queries the Windows device to verify that it has the required GPU capabilities and automatically reverts to server-side desktop composition if it does not. List the devices with the required GPU capabilities that do not meet the processor speed or RAM specifications in the GPO group for devices excluded from Desktop Composition Redirection.

The minimum available bandwidth is 1.5 Mbps; recommended bandwidth is 5 Mbps. Those values incorporate end-to-end latency.

The following clients are supported for Windows Media client-side content fetching, Windows Media redirection, and real-time Windows Media multimedia transcoding: Receiver for Windows, Receiver for iOS, and Receiver for Linux.

To use Windows Media client-side content fetching on Windows 8 devices, set the Citrix Multimedia Redirector as a default program: in Control Panel > Programs > Default Programs > Set your default programs, select Citrix Multimedia Redirector and click either Set this program as default or Choose defaults for this program.

GPU transcoding requires an NVIDIA CUDA-enabled GPU with Compute Capability 1.1 or higher; see <http://developer.nvidia.com/cuda/cuda-gpus>.

The following clients and Adobe Flash Players are supported:

- Receiver for Windows (for second generation Flash Redirection features) - Second generation Flash Redirection features require Adobe Flash Player for Other Browsers, sometimes referred to as an NPAPI (Netscape Plugin Application Programming Interface) Flash Player
- Receiver for Linux (for second generation Flash Redirection features) - Second generation Flash Redirection features require Adobe Flash Player for other Linux or Adobe Flash Player for Ubuntu.
- Citrix Online plug-in 12.1 (for legacy Flash Redirection features) - Legacy Flash Redirection features require Adobe Flash Player for Windows Internet Explorer (sometimes referred to as an ActiveX player).

The major version number of the Flash Player on the user device must be greater than or equal to the major version number of the Flash Player on the server. If an earlier version of the Flash Player is installed on the user device, or if the Flash Player cannot be installed on the user device, Flash content is rendered on the server.

The machines running VDAs require:

- Adobe Flash Player for Windows Internet Explorer (the ActiveX player)
- Internet Explorer versions 7, 8, 9, 10, 11 (in non-Modern UI mode). Flash redirection requires Internet Explorer on the server; with other browsers, Flash content is rendered on the server.
- Protected mode disabled in Internet Explorer (Tools > Internet Options > Security tab > Enable Protected Mode check box cleared). Restart Internet Explorer to effect the change.

When installing a VDA for Windows Desktop OS, you can choose to install the HDX 3D Pro version.

The physical or virtual machine hosting the application can use GPU Passthrough or Virtual GPU (vGPU):

- GPU Passthrough is available with Citrix XenServer. GPU Passthrough is also available with VMware vSphere and VMware ESX, where it is referred to as virtual Direct Graphics Acceleration (VDGA).

- vGPU is available with Citrix XenServer; see www.citrix.com/go/vGPU (Citrix My Account credentials required).

Citrix recommends that the host computer have at least 4 GB of RAM and four virtual CPUs with a clock speed of 2.3 GHz or higher.

Graphical Processing Unit (GPU):

- For CPU-based compression (including lossless compression), HDX 3D Pro supports any display adapter on the host computer that is compatible with the application being delivered.
- For optimized GPU frame buffer access using the NVIDIA GRID API, HDX 3D Pro requires NVIDIA Quadro cards with the latest NVIDIA drivers. The NVIDIA GRID delivers a high frame rate, resulting in a highly interactive user experience.
- For vGPU using XenServer, HDX 3D Pro requirements include NVIDIA GRID K1 and K2 cards.

User device:

- HDX 3D Pro supports all monitor resolutions that are supported by the GPU on the host computer. However, for optimum performance with the minimum recommended user device and GPU specifications, Citrix recommends a maximum monitor resolution for user devices of 1920 x 1200 pixels for LAN connections, and 1280 x 1024 pixels for WAN connections.
- Citrix recommends that user devices have at least 1 GB of RAM and a CPU with a clock speed of 1.6 GHz or higher. Use of the default deep compression codec, which is required on low-bandwidth connections, requires a more powerful CPU unless the decoding is done in hardware. For optimum performance, Citrix recommends that user devices have at least 2 GB of RAM and a dual-core CPU with a clock speed of 3 GHz or higher.
- For multi-monitor access, Citrix recommends user devices with quad-core CPUs.
- User devices do not need a dedicated GPU to access desktops or applications delivered with HDX 3D Pro.
- Citrix Receiver must be installed.

Supported clients: Citrix Receiver for Windows, Receiver for Mac, and Receiver for Linux.

Supported video conferencing applications:

- Citrix GoToMeeting HDFaces
- Adobe Connect
- Cisco WebEx
- IBM Sametime
- Microsoft Lync 2010 and 2013
- Microsoft Office Communicator
- Google+ Hangouts
- Media Foundation-based video applications on Windows 8.x, Windows Server 2012, and Windows Server 2012 R2
- Skype 6.7. To use Skype on a Windows client, edit the registry on the client and the server:
 - Client registry key HKEY_CURRENT_USER\Software\Citrix\HdxRealTime
 - Name: DefaultHeight , Type: REG_DWORD, Data: 240
 - Name: DefaultWidth, Type: REG_DWORD, Data: 320
 - Server registry key HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\Vd3d\Compatibility
 - Name: skype.exe, Type: REG_DWORD, Data: Set to 0

Other user device requirements:

- Appropriate hardware to produce sound.
- DirectShow-compatible webcam (use the webcam default settings). Webcams that are hardware encoding-capable

reduces client-side CPU usage.

- Webcam drivers, obtained from the camera manufacturer if possible.

UDP audio for Multi-Stream ICA is supported on Receiver for Windows and Receiver for Linux 13.

Echo cancellation is supported on Citrix Receiver for Windows.

- **Universal Print Server** - The Universal Print Server comprises client and server components. The UPClient component is included in the VDA installation. The UPServer component (which you install on each print server where the shared printers reside that you want to provision with the Citrix Universal Print Driver in user sessions) is supported on:
 - Windows Server 2008 R2 SP1
 - Windows Server 2008 32-bit
 - Windows Server 2012 R2 and 2012. This applies to minimum XenApp and XenDesktop 7.6 FP3.
- **XenApp and XenDesktop 7.6 FP3**. The following are prerequisites for installing the UPServer component on the print server:
 - Microsoft Visual Studio 2013 Runtime (both 32-bit and 64-bit)
 - Microsoft .NET Framework 4.5.1
 - CDF_x64.msi for 64-bit platforms or CDF_x86.msi for 32-bit platforms
 - UpsServer_x64.msi for 64-bit platforms or UpsServer_x86.msi for 32-bit platforms

The UPClient component, which you install on XenApp and XenDesktop hosts that provision session network printers, is compatible with Windows 10 desktops and is part of the VDA installation.

User authentication during printing operations requires the Universal Print Server to be joined to the same domain as the Remote Desktop Services VDA.

- Citrix recommends installing or upgrading to the component software versions provided on the installation media for this release.
 - StoreFront requires 2 GB of memory. See the StoreFront documentation for system requirements. StoreFront 2.6 is the minimum supported version with this release.
 - When using Provisioning Services with this release, the minimum supported Provisioning Services version is 7.0.
 - The Citrix License Server requires 40 MB of disk space. See the licensing documentation for system requirements. Only Citrix License Server for Windows is supported. The minimum supported version is 11.12.1.
- The Microsoft Group Policy Management Console (GPMC) is required if you store Citrix policy information in Active Directory rather than the Site Configuration database. For more information, see the Microsoft documentation.
- By default, the Receiver for Windows is installed when you install a VDA. For system requirements information on other platforms, see the Receiver for Windows documentation.
- The Receiver for Linux and the Receiver for Mac are provided on the product installation media. See their documentation for system requirements.
- When using Access Gateway versions earlier than 10.0 with this release, Windows 8.1 and Windows 8 clients are not supported.
- **Desktop Lock** - Supported operating systems:
 - Windows 7, including Embedded Edition
 - Windows XP Embedded

- Windows Vista

User devices must be connected to a local area network (LAN).

Supported Receiver: Citrix Receiver for Windows Enterprise 3.4 package (minimum).

- Client folder redirection - Supported operating systems:
 - Server: Windows Server 2008 R2 SP1, Windows Server 2012, and Windows Server 2012 R2
 - Client (with latest Citrix Receiver for Windows): Windows 7, Windows 8, and Windows 8.1
- Multiple network interface cards are supported.
- See the [App-V](#) article for supported versions.

技术概述

May 04, 2016

XenApp 和 XenDesktop 是虚拟化解决方案。利用这些方案，IT 可以在提供随时随地访问任何设备的同时，控制虚拟机、应用程序、许可和安全性。

XenApp 和 XenDesktop 允许：

- 最终用户独立于设备的操作系统和界面运行应用程序和桌面。
- 管理员管理网络并提供或限制对选定设备或全部设备的访问。
- 管理员从单个数据中心管理整个网络。

XenApp 和 XenDesktop 共享统一的体系结构 FlexCast Management Architecture (FMA)。FMA 的关键功能是通过单个站点和集成置备运行多个版本的 XenApp 或 XenDesktop。

典型的 XenApp 或 XenDesktop 环境包括几项关键技术组件。这些组件在用户连接到应用程序和桌面时相互作用，并记录有关站点活动的数据。

Citrix Receiver

一款安装在用户设备上的软件客户端，通过 TCP 端口 80 或 443 提供与虚拟机的连接，并使用 StoreFront Service API 与 StoreFront 通信。

StoreFront

用于对用户进行身份验证、管理应用程序和桌面并托管应用程序存储的接口。StoreFront 使用 XML 与 Delivery Controller 通信。

Delivery Controller

XenApp 或 XenDesktop 站点的中央管理组件，由用于管理资源、应用程序和桌面的服务组成，负责优化和平衡用户连接的负载。

Virtual Delivery Agent (VDA)

安装在运行 Windows 服务器或 Windows 桌面操作系统的计算机上的代理，允许这些计算机及其所托管的资源供用户使用。在运行 Windows 服务器操作系统的计算机上安装的 VDA 允许此计算机托管多个用户的多个连接，并通过以下其中一个端口连接到用户：

- 如果启用 SSL：TCP 端口 80 或端口 443
- 如果启用了通用网关协议 (CGP)（用于启用会话可靠性）：TCP 端口 2598
- 如果禁用 CGP 或用户使用旧版客户端进行连接：TCP 端口 1494

Broker Service

一种 Delivery Controller 服务，用于跟踪登录的用户和登录位置、用户拥有的会话资源以及用户是否需要重新连接到现有应用程序。Broker Service 执行 PowerShell 并通过 TCP 端口 80 与 Broker Agent 通信。它不可以使用 TCP 端口 443。

Broker Agent

托管多个插件并收集实时数据的代理。Broker Agent 位于 VDA 上并通过 TCP 端口 80 连接到 Controller。它不可以使用 TCP 端口 443。

监视服务

一种 Delivery Controller 组件，用于收集历史数据并在默认情况下将其存放在站点数据库中。监视服务通过 TCP 端口 80 或 443 通信。

ICA 文件/堆栈

捆绑的用户信息，在连接到 VDA 时需要此信息。

站点数据库

Microsoft SQL 数据库，用于存储 Delivery Controller 的数据，如站点策略、计算机目录和交付组。

NetScaler Gateway

数据访问解决方案，使用其他凭据在 LAN 防火墙内外提供安全访问。

Director

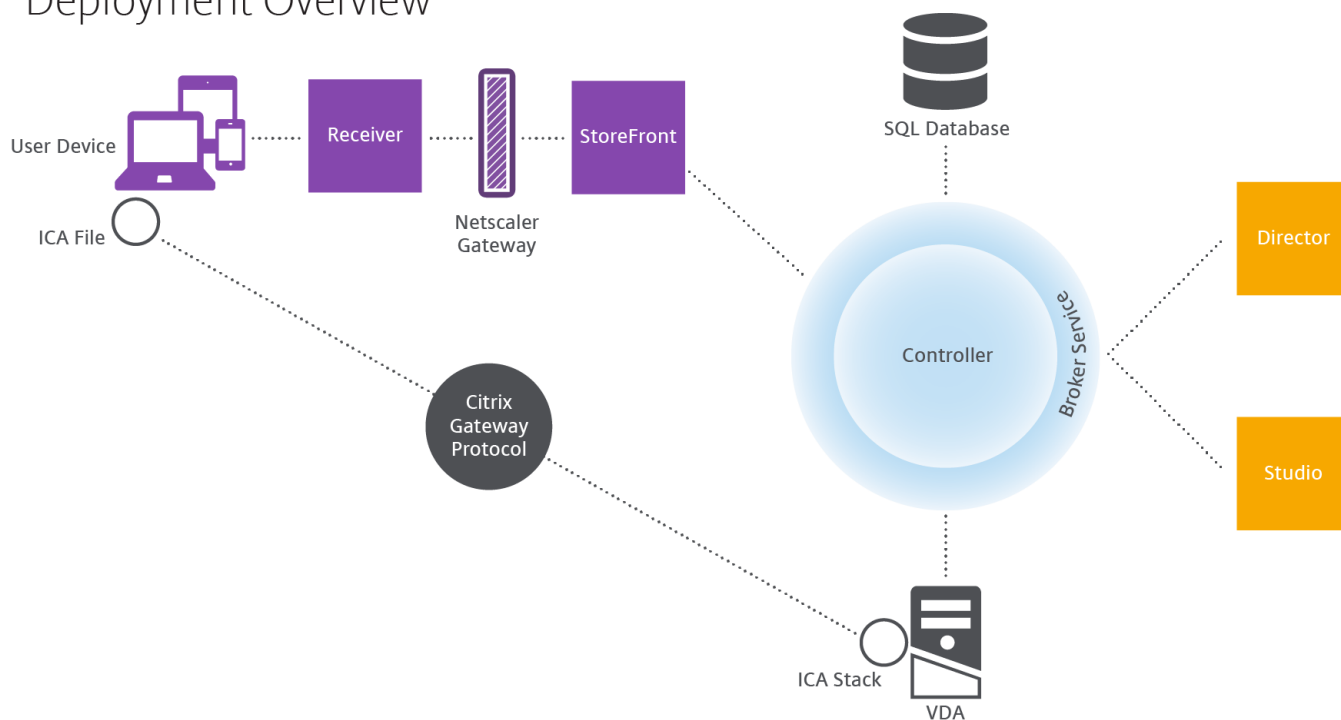
基于 Web 的工具，允许管理员访问 Broker Agent 的实时数据、站点数据库中的历史数据和 NetScaler 中的 HDX 数据，以进行故障排除和提供支持。Director 通过 TCP 端口 80 或 443 与 Controller 通信。

Studio

管理控制台，允许管理员配置和管理站点，授予对 Broker Agent 中的实时数据的访问权限。Studio 通过 TCP 端口 80 与 Controller 通信。

XenApp 和 XenDesktop 站点由具有专用角色的计算机组成，用于实现可扩展性、高可用性和故障转移，并提供设计安全的解决方案。XenApp 或 XenDesktop 站点包括安装 VDA 的 Windows 服务器和桌面计算机，以及用于管理访问权限的 Delivery Controller。

Deployment Overview



VDA 使用户能够连接到桌面和应用程序。它安装在数据中心内的服务器或桌面计算机上以实现大多数交付方法，但是也可以安装在物理 PC 上以实现 Remote PC Access。

Controller 由独立的 Windows 服务组成，用于管理资源、应用程序和桌面，并优化和平衡用户连接。每个站点具有一个或多个 Controller，并且，由于会话依赖延迟、带宽和网络可靠性，在理想状态下，所有 Controller 应该位于相同 LAN 上。

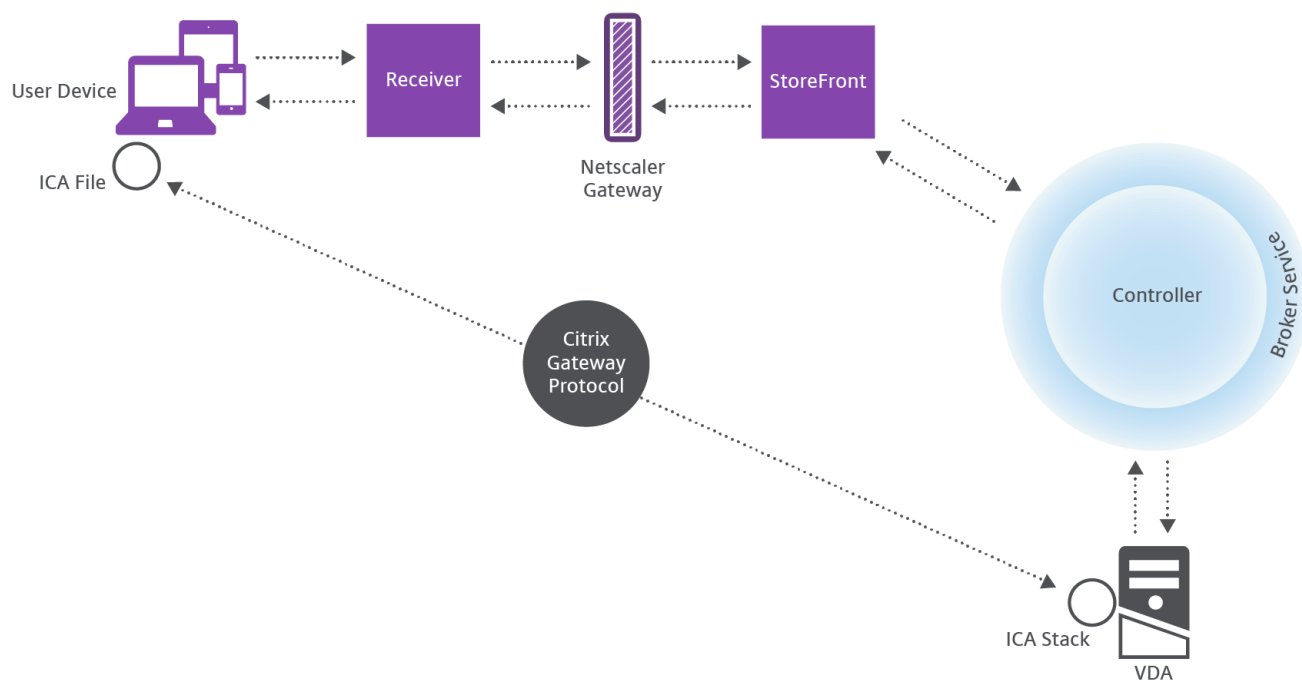
用户绝对不能直接访问 Controller。VDA 充当用户和 Controller 之间的媒介。当用户使用 StoreFront 登录到站点时，其凭据被传递到 Broker Service。Broker Service 根据为用户设置的策略获取其配置文件和可用资源。

要启动 XenApp 或 XenDesktop 会话，用户通过 Citrix Receiver（安装在用户设备上）或 Receiver for Web (RFW) 进行连接。

在 Receiver 内部，用户选择所需的物理或虚拟桌面或虚拟应用程序。

用户的凭据按照此路径进行传递以访问 Controller，后者通过与 Broker Service 通信确定所需的资源。建议管理员在 StoreFront 上放置一个 SSL 证书以加密来自 Receiver 的凭据。

User connections



Broker Service 决定允许用户访问的桌面和应用程序。

验证凭据后，有关可用应用程序或桌面的信息通过 StoreFront-Receiver 路径发送给用户。用户选择此列表中的应用程序或桌面时，该信息按照相反路径返回到 Controller，Controller 决定托管特定应用程序或桌面的 VDA。

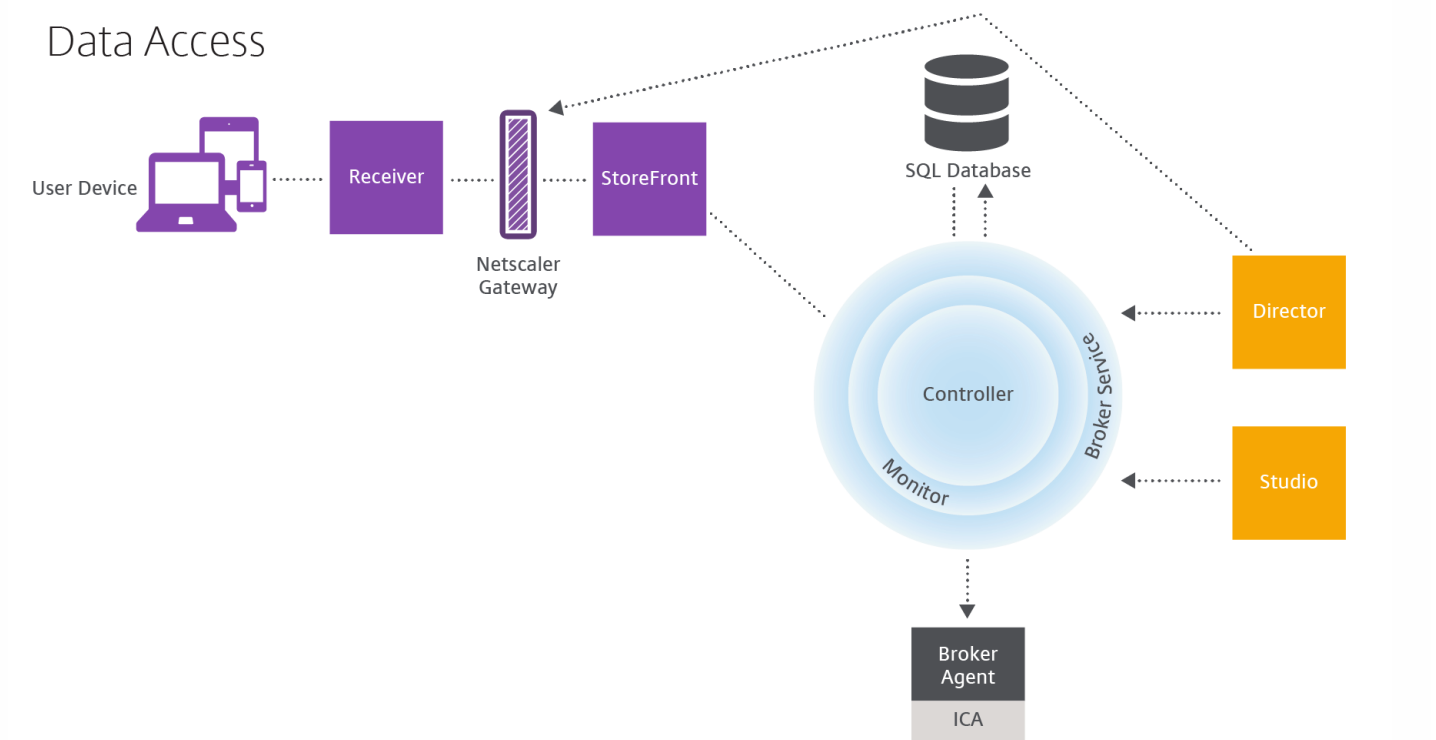
Controller 将用户的凭据通过消息发送给 VDA，并将关于用户和连接的所有数据发送给 VDA。VDA 接受连接，并将该信息按原路径返回给 Receiver。Receiver 将会话中已生成的所有信息绑定在一起，从而在用户设备上（如果 Receiver 安装在本地）或 RFW 上（如果通过 Web 访问）创建 Independent Computing Architecture (ICA) 文件。只要站点经过正确设置，凭据在整个流程均保留加密状态。

ICA 文件被复制到用户设备上，并在设备与 VDA 上运行的 ICA 堆栈之间建立直接连接。此连接绕过管理基础结构：Receiver、StoreFront 和 Controller。

Receiver 和 VDA 之间的连接使用 Citrix Gateway Protocol (CGP)。如果连接丢失，通过会话可靠性功能，用户可以重新连接到 VDA，而无需通过管理基础结构重新启动。可以在 Studio 中启用或禁用会话可靠性。

客户端连接到 VDA 后，VDA 会通知 Controller 用户已登录，Controller 将此信息发送到站点数据库，并开始在监视数据库中记录数据。

IT 可以通过 Studio 或 Director 访问每个 XenApp 或 XenDesktop 会话生成的数据。Studio 允许管理员访问 Broker Agent 中的实时数据，以便更好地管理站点。Director 可以访问此实时数据、存储在监视数据库中的历史数据以及 NetScaler Gateway 中的 HDX 数据，用于提供技术支持和故障排除。



在 Controller 内部，Broker Service 报告虚拟机上的每个会话的会话数据，以提供实时数据。监视服务还跟踪实时数据并将其作为历史数据存储在监视数据库中。

Studio 只能与 Broker Service 通信，因此也只能访问实时数据。Director 可以与 Broker Service 通信（通过 Broker Agent 中的插件）以访问站点数据库。

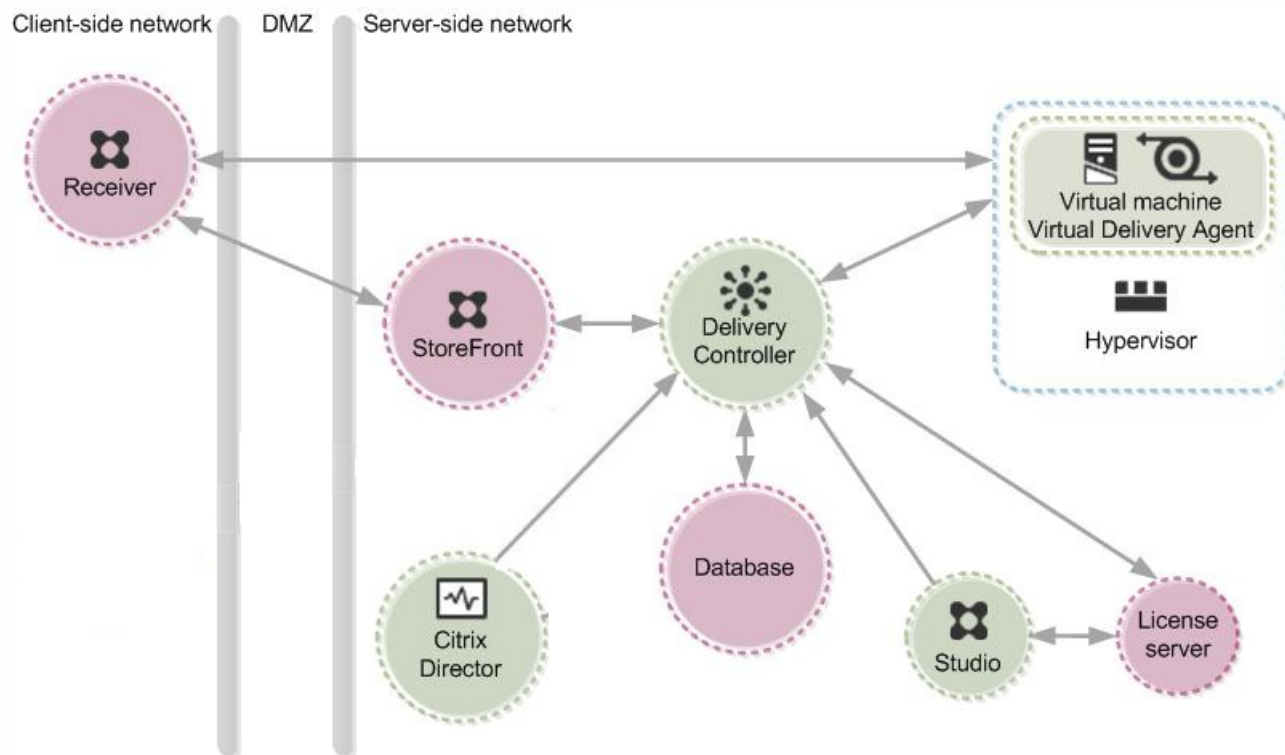
Director 还可以访问 NetScaler Gateway 以获取 HDX 数据信息。

- [概念和组件](#)
- [Active Directory](#)
- [容错](#)
- [交付方法](#)

概念和组件

Jul 13, 2016

此图显示了典型 XenApp 或 XenDesktop 部署（称为站点）的主要组件。



此图中的组件包括：

- **Delivery Controller** — Delivery Controller 是任何 XenApp 或 XenDesktop 站点的中心管理组件。每个站点有一个或多个 Delivery Controller。至少安装在数据中心内的一个服务器上。（为实现站点可靠性和可用性，将 Controller 安装在多个服务器上。）Controller 由多项服务组成，这些服务通过与虚拟机管理程序通信来分发应用程序和桌面、对用户进行身份验证并管理用户访问、代理用户与其虚拟桌面和应用程序之间的连接、优化使用连接并对这些连接进行负载平衡。各个服务的数据存储在站点数据库中。

Controller 管理桌面的状态，并根据需要和管理配置启动/停止桌面。在某些版本中，Controller 允许您安装 Profile Management 以在虚拟化或物理 Windows 环境中管理用户个性化设置。

- **数据库** — 每个 XenApp 或 XenDesktop 站点至少具有一个 Microsoft SQL Server 数据库，用于存储所有配置信息和会话信息。此数据库存储组成 Controller 的服务所收集并管理的数据。在数据中心内安装此数据库，并确保此数据库与 Controller 建立永久连接。
- **Virtual Delivery Agent (VDA)** — VDA 安装在站点中要供用户使用的各个物理计算机或虚拟机上。它使计算机向 Controller 注册，Controller 进而允许向用户提供它所托管的计算机和资源。VDA 建立并管理计算机与用户设备之间的连接，确认 Citrix 许可证可供用户或会话使用，并应用已为会话配置的任何策略。VDA 通过 VDA 中包含的 Broker Agent 将会话信息传递给 Controller 中的 Broker Service。
XenApp 和 XenDesktop 包含适用于 Windows 服务器和桌面操作系统的 VDA。适用于 Windows 服务器操作系统的 VDA 允许多个用户同时连接到服务器。适用于 Windows 桌面的 VDA 每次仅允许一个用户连接到桌面。

- **StoreFront** — StoreFront 可对托管资源的站点的用户进行身份验证，并可管理用户访问的桌面和应用程序的存储。它托管企业应用商店，使用户可以自助访问您为其提供的桌面和应用程序。它还跟踪用户的应用程序订阅、快捷方式名称和其他数据，以确保在多个设备之间向用户提供一致的体验。
- **Receiver** — 安装在用户设备和其他端点（如虚拟桌面）上。Citrix Receiver 可使用户能够快速安全地从任何用户设备（包括智能手机、平板电脑和 PC）自助访问文档、应用程序和桌面。通过 Receiver 可以对 Windows、Web 和软件即服务 (SaaS) 应用程序进行按需访问。对于无法安装 Receiver 软件的设备，Receiver for HTML5 通过与 HTML5 兼容的 Web 浏览器提供连接。
- **Studio** — 是可用于配置和管理 XenDesktop 部署的管理控制台，使使用此控制台则不需要为管理应用程序和桌面的交付而配置单独的管理控制台。Studio 提供多种向导来引导您完成设置环境、创建用于托管应用程序和桌面的工作负载并将应用程序和桌面分配给用户的过程。还可以使用 Studio 为站点分配和跟踪 Citrix 许可证。
Studio 从 Controller 中的 Broker Service 获取所显示的信息。

- **Director** — Director 是一款基于 Web 的工具，IT 支持团队和技术支持团队可以利用该工具监控环境和进行故障排除，以避免这些问题危及系统。此外，还可以为最终用户执行支持任务。可以使用一个 Director 部署连接和监视多个 XenApp 或 XenDesktop 站点。

Director 显示以下来源的会话和站点信息：

- 来自 Controller 中的 Broker Service 的实时会话数据，其中包括 Broker Service 从 VDA 中的 Broker Agent 获取的数据。
- 来自 Controller 中的 Monitor Service 的历史站点数据。
- HDX Insight 从 NetScaler 捕获的有关 HDX 通信（也称为 ICA 通信）的数据，前提是部署中包含 NetScaler，并且 XenApp 或 XenDesktop 版本包含 HDX Insight。

通过使用 Microsoft 远程协助，还可以查看用户会话并与之交互。

- **许可证服务器** — 许可证服务器用于管理产品许可证。它与 Controller 通信以管理每个用户会话的许可，与 Studio 通信以分配许可证文件。必须至少创建一个许可证服务器来存储和管理许可证文件。
- **虚拟机管理程序** — 虚拟机管理程序托管站点中的虚拟机。这些虚拟机可以是用于托管应用程序和桌面的虚拟机，也可以是用于托管 XenApp 和 XenDesktop 组件的虚拟机。虚拟机管理程序安装在完全专用于运行虚拟机管理程序和托管虚拟机的主机计算机上。

Citrix XenServer 虚拟机管理程序包含在 XenApp 和 XenDesktop 中，但是您也可以使用其他受支持的虚拟机管理程序，如 Microsoft Hyper-V 或 VMware vSphere。

尽管大多数 XenApp 和 XenDesktop 实现需要一个虚拟机管理程序，但您不需要使用虚拟机管理程序来提供 Remote PC Access，或者在使用 Provisioning Services（包含在某些 XenApp 和 XenDesktop 版本中）而非 MCS 时来置备虚拟机。

XenApp 或 XenDesktop 部署中也可能包含以下其他组件（未显示在上面的插图中）：

- **Provisioning Services** — Provisioning Services 是 XenApp 和 XenDesktop 的可选组件，在有些版本中提供。它是 MCS 的备选方式，用于置备虚拟机。MCS 创建主映像的副本，Provisioning Services 采用流技术将主映像推送到用户设备。Provisioning Services 执行此操作时无需使用虚拟机管理程序，因此，您可以使用它来托管物理计算机。站点中包含 Provisioning Services 时，Provisioning Services 与 Controller 通信以向用户提供资源。
- **NetScaler Gateway** — 用户从公司防火墙外部连接时，此版本可以使用 Citrix NetScaler Gateway（以前称为 Access Gateway）技术保护与 SSL 连接时的安全性。NetScaler Gateway 或 NetScaler VPX 虚拟设备是在隔离区域 (DMZ) 中部署的 SSL VPN 设备，用于通过公司防火墙提供单个安全访问点。
- **Citrix CloudBridge** — 在向位于远程位置（如分支机构）的用户交付虚拟桌面的部署中，可以采用 Citrix CloudBridge（以前称为 Citrix Branch Repeater 或 WANScaler）技术来优化性能。Repeater 可提高整个广域网的性能，因此通过在网络中使用 Repeater，分支机构用户将在 WAN 上体验到像 LAN 一般的性能。例如，CloudBridge 可以设置用户体验不同部分的优先级，以便通过网络发送大文件或打印作业时，分支机构的用户体验不会降低。带有 CloudBridge 的 HDX WAN 优化提供令牌索引化压缩和重复数据删除功能，极大地降低了带宽要求并改进了性能。有关详细信息，请参阅 Citrix CloudBridge 文档。

利用 XenApp 和 XenDesktop，您可以通过计算机目录设置希望向用户提供的资源，但是，通过交付组指定可以访问这些资源的用户。

计算机目录

计算机目录是作为单个实体进行管理的虚拟机或物理机集合。 这些计算机及其上的应用程序或虚拟桌面是要提供给用户的资源。 计算机目录中的所有计算机安装相同的操作系统和 VDA，并且，这些计算机上提供相同的应用程序或虚拟桌面。 通常，您创建一个主映像，然后使用此主映像在目录中创建完全相同的虚拟机。

创建计算机目录时，为目录中的计算机指定计算机的类型和置备方法。

计算机类型

- Windows 服务器操作系统计算机 — 基于 Windows 服务器操作系统的虚拟机或物理计算机，用于交付 XenApp 发布的应用程序（也称为基于服务器的托管应用程序）和 XenApp 发布的桌面（也称为服务器托管的桌面）。 这些计算机允许多个用户同时与其建立连接。
- 桌面操作系统计算机 — 基于 Windows 桌面操作系统的虚拟机或物理计算机，用于交付 VDI 桌面（运行 Window 桌面操作系统并且可以完全进行个性化设置的桌面，具体取决于您选择的选项）、VM 托管应用程序（来自桌面操作系统的应用程序）和托管的物理桌面。 同一时间仅允许一个用户与其中的一个计算机建立连接。
- Remote PC Access — 白名单中包括的用户设备，这些设备使用户可以从任何运行 Citrix Receiver 的设备远程访问其办公室 PC 上的资源。 Remote PC Access 使您可以通过 XenDesktop 部署管理对脱机 PC 的访问。

置备方法

- Machine Creation Services (MCS) — 一组服务，可以根据需要从主映像创建虚拟服务器和桌面、优化存储利用率并在用户每次登录时为其提供虚拟机。 Machine Creation Services 完全集成于 Citrix Studio 中，并在其中进行管理。
- Provisioning Services — 允许从单个共享磁盘映像实时置备或重新置备计算机。 Provisioning Services 将目标设备作为一个设备集合进行管理。 桌面和应用程序通过从主目标设备获得映像的 Provisioning Services 虚拟磁盘进行交付，并使您能够利用物理硬件或虚拟机的处理能力。 Provisioning Services 通过其自身的控制台进行管理。
- 现有映像 — 适用于已迁移到数据中心内的虚拟机的桌面和应用程序。 必须单独或统一使用第三方电子软件分发 (ESD) 工具管理目标设备。

交付组

交付组是用户的集合，这些用户具有通用资源组的访问权限。 交付组包含计算机目录中的计算机和具有站点访问权限的 Active Directory 用户。 由于 Active Directory 组和交付组均是将具有相似要求的用户分成一组，因此，通常按照用户所属的 Active Directory 组将其分配到交付组。

每个交付组可以包含多个计算机目录中的计算机，每个计算机目录可以向多个交付组提供计算机，但是每个单独的计算机在同一时间只能属于一个交付组。 可以将交付组设置为交付应用程序、桌面或同时交付二者。

可以定义交付组中的用户可以访问的资源。 例如，如果要向不同的用户交付不同的应用程序，一种方式是在一个计算机目录的主映像上安装要交付的所有应用程序，并在此目录中创建足够多的计算机以在多个交付组之间分发。 然后，配置各个交付组，以交付计算机上安装的不同应用程序子集。

如果您熟悉 XenApp 6.5 和 XenApp 的早期版本，可能会有助于您了解 XenApp 7.6 和 XenDesktop 7.6 与这些版本的不同之处。

尽管并不完全相同，但以下表格可帮助您将 XenApp 6.5 和早期版本中的功能元素与 XenApp 7.6 和 XenDesktop 7.6 中的功能元素对应起来：

XenApp 6.5 和更早版本中被代替的元素：	XenApp 和 XenDesktop 7.6 中使用的元素：

Independent Management Architecture (IMA) XenApp 6.5 和更早版本中被代替的元素：	FlexCast Management Architecture (FMA) XenApp 和 XenDesktop 7.6 中使用的元素：
场	站点
工作组	计算机目录 交付组
工作进程	Virtual Delivery Agent (VDA) 服务器操作系统计算机，服务器操作系统 VDA 桌面操作系统计算机，桌面操作系统 VDA
远程桌面服务 (RDS) 或终端服务计算机	服务器操作系统计算机，服务器操作系统 VDA
区域和数据收集器	Delivery Controller
交付服务控制台	Citrix Studio 和 Citrix Director
发布应用程序	交付应用程序
数据存储	数据库
负载评估程序	负载管理策略
管理员	委派的管理员 角色 作用域

XenApp 7.6 和 XenDesktop 7.6 基于 FlexCast Management Architecture (FMA)。FMA 是面向服务的体系结构，可以实现跨 Citrix 各种技术的互操作性和模块化管理。FMA 为应用程序交付、移动性、服务、灵活置备和云管理提供了一个统一的平台。

FMA 取代了 XenApp 6.5 和早期版本中使用的 Independent Management Architecture (IMA)。

在考虑与 XenApp 6.5 和早期版本中的元素的关系时，以下是 FMA 中的关键元素：

交付站点

场是 XenApp 6.5 和早期版本中的顶层对象。在 XenApp 7.6 和 XenDesktop 7.6 中，交付站点则是级别最高的项目。站点为用户组提供应用程序和桌面。

FMA 则要求您必须位于域中才能部署站点。例如，要安装服务器，您的帐户必须具有本地管理员权限并且是 Active Directory 中的域用户。

计算机目录和交付组

XenApp 6.5 和早期版本中托管应用程序的计算机属于工作组，目的是便于有效地管理应用程序和服务器软件。管理员可以将一个工作组中的所有计算机作为一个单元进行管理，以满足其应用程序管理和负载平衡需求。使用文件夹来组织应用程序和计算机。

在 XenApp 7.6 和 XenDesktop 7.6 中，结合使用计算机目录和交付组来管理计算机、负载平衡和托管应用程序或桌面。

Virtual Delivery Agent

在 XenApp 6.5 和早期版本中，工作组中的工作计算机运行用户的应用程序并与数据收集器通信。在 XenApp 7.6 和 XenDesktop 7.6 中，则是 VDA 与管理用户连接的 Delivery Controller 通信。

Delivery Controller

在 XenApp 6.5 和早期版本中，区域主服务器负责处理用户连接请求以及与虚拟机管理程序的通信。在 XenApp 7.6 和 XenDesktop 7.6 中，站点中的 Controller 负责分发和处理连接请求。

在 XenApp 6.5 和早期版本中，区域提供了一种跨 WAN 连接聚合服务器和重复数据的方式。尽管在 XenApp 7.6 和 XenDesktop 7.6 中没有区域的确切等效对象，但您可以为用户提供跨多个 WAN 和位置的应用程序。您可以为特定的地理位置或数据中心设计交付站点，然后允许您的用户访问多个交付站点。XenApp 7.6 和 XenDesktop 7.6 的 App Orchestration 提供管理位于多个地理位置的多个站点功能。

Citrix Studio 和 Citrix Director

使用 Studio 控制台配置环境并为用户提供对应用程序和桌面的访问权限。Studio 取代了 XenApp 6.5 和早期版本中的交付服务控制台。

管理员使用 Director 监视环境、重影用户设备和对 IT 问题进行故障排除。要重影用户，必须启用 Microsoft 远程协助；安装 VDA 时默认启用此功能。

交付应用程序

XenApp 6.5 和早期版本使用发布应用程序向导来准备应用程序并将其交付给用户。在 XenApp 7.6 和 XenDesktop 7.6 中，使用 Studio 创建和添加应用程序，使其可供交付组中的用户使用。在使用 Studio 时，首先配置站点，创建并指定计算机目录，然后在这些计算机目录中创建交付组。交付组确定哪些用户可以访问您交付的应用程序。

数据库

XenApp 7.6 和 XenDesktop 7.6 不使用 IMA 数据存储来存储配置信息，而是使用 Microsoft SQL Server 数据库来存储配置和会话信息。

负载管理策略

在 XenApp 6.5 和早期版本中，负载评估程序使用预定义的衡量指标来确定计算机上的负载。用户连接可以匹配到负载更少的计算机。

在 XenApp 7.6 和 XenDesktop 7.6 中，则使用负载管理策略跨多个计算机平衡负载。

委派的管理员

在 XenApp 6.5 和早期版本中，创建自定义管理员并根据文件夹和对象向其分配权限。在 XenApp 7.6 和 XenDesktop 7.6 中，则基于角色和作用域来创建自定义管理员。角色表示一种作业职能，并且具有定义的关联权限以允许委派。作用域表示对象集合。内置管理员角色具有特定的权限集，如技术支持、应用程序、托管和目录。例如，技术支持管理员只能与指定站点上的各个用户协作，而完全权限管理员可以监视整个部署并解决整个系统范围的 IT 问题。

过渡到 FMA 意味着在 XenApp 6.5 和早期版本中提供的一些功能可能会采用其他方式实现，或者可能需要替换其他功能、组件或工具才能实现相同的目的。

XenApp 6.5 和更早版本中被代替的元素：	XenApp 和 XenDesktop 7.6 中使用的元素：
使用策略设置配置会话预启动和会话延迟	通过编辑交付组设置配置会话预启动和会话延迟。 与 XenApp 6.5 中相同，这些功能通过以下方式帮助用户快速连接到应用程序：在用户请求会话之前启动会话（会话预启动），并在用户关闭所有应用程序之后使会话保持活动状态（会

XenApp 6.5 和更早版本中被代替的元素：	话延迟)。在 XenApp 和 XenDesktop 7.6 中，通过为现有交付组配置这些设置为指定用户启用这些功能。请参阅 配置会话预启动和会话延迟 。
通过在设置已发布应用程序的属性时向匿名用户授予权限来提供对未经身份验证（匿名）用户的支持	通过在设置交付组的用户属性时配置此选项提供对未经身份验证（匿名）用户的支持。请参阅 用户 。
即使在与数据存储的连接不可用时，本地主机缓存仍允许工作服务器正常运行	连接租用使用户可以连接和重新连接到他们最后使用的应用程序和桌面，即使站点数据库不可用也是如此。连接租用功能是对 SQL Server 高可用性最佳实践的补充。请参阅 连接租用 。
应用程序流技术推送	App-V 提供流应用程序，并使用 Studio 进行管理。
Web Interface	Citrix 建议过渡到 StoreFront。
SmartAuditor	使用配置日志记录从管理角度记录所有会话活动，或使用兼容 Citrix 的第三方工具记录会话。

Active Directory

May 04, 2016

进行身份验证和授权时需要使用 Active Directory。Active Directory 中的 Kerberos 基础结构用于保证与 Delivery Controller 通信的真实性和保密性。有关 Kerberos 的详细信息，请参阅 Microsoft 文档。

[系统要求](#)文档列出了支持的林和域功能级别。要使用策略建模，域控制器必须在 Windows Server 2003 到 Windows Server 2012 R2 上运行；这不会影响域功能级别。

本产品支持：

- 具有以下特征的部署：用户帐户和计算机帐户所在的域位于同一 Active Directory 林中。用户和计算机帐户可以存在于同一林中的任意域内。所有域功能级别和林功能级别在这种类型的部署中都得到支持。
- 具有以下特征的部署：用户帐户所在的 Active Directory 林不同于控制器和虚拟桌面的计算机帐户所在的 Active Directory 林。在此类部署中，包含控制器和虚拟桌面计算机帐户的域必须信任包含用户帐户的域。可以使用林信任和外部信任。所有域功能级别和林功能级别在这种类型的部署中都得到支持。
- 具有以下特征的部署：在该部署中，控制器的计算机帐户所在的 Active Directory 林不同于虚拟桌面的计算机帐户所在的一个或多个附加 Active Directory 林。在此类部署中，在控制器计算机帐户所在的域与虚拟桌面计算机帐户所在的所有域之间必须存在双向信任关系。在此类部署中，包含控制器或虚拟桌面计算机帐户的所有域都必须处于“Windows 2000 本机”功能级别或更高级别。所有林功能级别都得到支持。
- 可写域控制器。不支持只读域控制器。

或者，Virtual Delivery Agent (VDA) 可以使用在 Active Directory 中发布的信息来确定可以注册的控制器（发现）。支持此方法的目的是实现向后兼容，并且此方法仅在 VDA 与控制器位于相同的 Active Directory 林中时可用。有关此发现方法的信息，请参阅[基于 Active Directory OU 的控制器发现](#)和 [CTX118976](#)。

注意：此信息适用的最低版本为 XenDesktop 7.1 和 XenApp 7.5，不适用于早期版本的 XenDesktop 或 XenApp。

在具有多个林的 Active Directory 环境中，如果已配置单向或双向信任，则可以使用 DNS 转发器执行名称查找和注册。要允许相应的 Active Directory 用户创建计算机帐户，请使用控制委派向导。有关此向导的详细信息，请参阅 Microsoft 文档。

如果已在两个林之间配置相应的 DNS 转发器，则不需要在 DNS 基础结构中配置反向 DNS 区域。

无论 Active Directory 和 NetBios 名称是否相同，如果 VDA 和 Controller 位于不同的林中，则需要创建

SupportMultipleForest 注册表项。只有 VDA 需要 SupportMultipleForest 注册表项。请使用以下信息添加注册表项：

警告：注册表编辑不当会导致严重问题，可能导致需要重新安装操作系统。Citrix 无法保证因“注册表编辑器”使用不当导致出现的问题能够得以解决。使用“注册表编辑器”需自担风险。在编辑注册表之前，请务必进行备份。

- HKEY_LOCAL_MACHINE\Software\Citrix\VirtualDesktopAgent\SupportMultipleForest
 - 名称：SupportMultipleForest
 - 类型：REG_DWORD
 - 数据：0x00000001 (1)

如果 DNS 命名空间与 Active Directory 的命名空间不同，您可能需要反向 DNS 配置。

如果设置期间已配置外部信任，则需要创建 ListOfSIDs 注册表项。如果 Active Directory FQDN 与 DNS FQDN 不同，或者如果包含域控制器的域具有的 Netbios 名称与 Active Directory FQDN 不同，也需要创建 ListOfSIDs 注册表项。要添加此注册表项，请使用以下信息：

- 对于 32 位或 64 位 VDA，请找到注册表项 HKEY_LOCAL_MACHINE\Software\Citrix\VirtualDesktopAgent\ListOfSIDs
 - 名称：ListOfSIDs

- 类型：REG_SZ
- 数据：控制器的安全标识符 (SID)

如果具有现有外部信任，应对 VDA 做以下更改：

1. 找到文件 \Citrix\Virtual Desktop Agent\brokeragentconfig.exe.config。
2. 备份该文件。
3. 在文本编辑程序（例如记事本）中打开该文件。
4. 找到文本 allowNtlm="false"，并将其更改为 allowNtlm="true"。
5. 保存该文件。

在添加 ListOfSIDs 注册表项并编辑 brokeragent.exe.config 文件之后，重新启动 Citrix Desktop Service 以应用所做的更改。

下表列出了支持的信任类型：

信任类型	传递性	方向	此版本支持
父与子	可传递	双向	是
树-根	可传递	双向	是
外部	不可传递	单向或双向	是
林	可传递	单向或双向	是
快捷方式	可传递	单向或双向	是
领域	可传递或非可传递	单向或双向	否

有关复杂 Active Directory 环境的详细信息，请参阅 [CTX134971](#)。

容错

May 04, 2016

本文档概述了可用来提高部署中容错级别的方式，从而确保业务关键型应用程序和桌面始终可用。

所有信息都存储在站点配置数据库中；Delivery Controller 只与数据库进行通信，彼此之间并不通信。拔出或关闭一个控制器不会对站点中的其他控制器产生影响。但是，这意味着站点配置数据库会形成单点故障。如果数据库服务器出现故障，除非用户从虚拟桌面注销或与虚拟桌面断开连接，否则与虚拟桌面的现有连接会继续工作。如果数据库服务器不可用，将无法建立新连接。

Citrix 建议您定期备份数据库，以便在数据库服务器出现故障时可以通过备份进行还原。另外，可以考虑采取几种高可用性解决方案以确保实现自动故障转移：

- SQL 镜像 — 建议采用此解决方案。通过镜像数据库可以确保一旦与活动数据库服务器失去联系，可以在几秒钟内实现自动故障转移，因此用户通常不会受到影响。但是，此方法比其他解决方案更为昂贵，因为在每台数据库服务器上必须使用完整的 SQL Server 许可证；在镜像环境中不能使用 SQL Server Express 版本。
- 使用虚拟机管理程序的高可用性功能 — 通过此方法，您可以将数据库作为虚拟机进行部署，并使用虚拟机管理程序的高可用性功能。此解决方案的成本比镜像方法要低，因为它使用的是现有虚拟机管理程序软件，您也可以使用 SQL Express。但是，其自动故障转移过程比较慢，因为需要花时间为数据库启动新计算机，这样可能会导致为用户提供的服务中断。
- SQL 群集化 — 可以使用 Microsoft 的 SQL 群集化技术，允许一台服务器自动接管另一台故障服务器的任务和职责。但是，该解决方案的设置更为复杂，自动故障转移过程通常比其他备选方案（如 SQL 镜像）更慢。
- AlwaysOn 可用性组是 SQL Server 2012 中引入的具有高可用性和灾难恢复能力的企业级解决方案。此方案可以使您最大程度地提高一个或多个用户数据库的可用性。AlwaysOn 可用性组要求 SQL Server 实例必须驻留在 Windows Server 故障转移群集 (WSFC) 节点上。有关详细信息，请参阅 [AlwaysOn 可用性组 \(SQL Server\)](#)。

注意：在 SQL 群集或 SQL 镜像安装中，不支持在节点上安装控制器。

在配置过程中，管理员需要在创建站点之前使用 SQL Server 管理工具完成多项任务。其余任务在管理员运行站点创建向导时执行。

镜像环境至少需要两个 SQL Server 计算机（在以下示例中为 SQL Server A 和 SQL Server B）。不可以使用 SQL Server Express Edition 作为主体数据库或镜像数据库。

通过 Microsoft SQL Server 管理工具，配置 SQL Server 数据库：

1. 在 SQL Server A 和 SQL Server B 上安装 SQL Server 软件。
2. 在 SQL Server A 上，创建要作为主体数据库的数据库（例如 myDatabaseMirror）。
 - 确保该数据库使用完全恢复模型而不是简单模型。（默认情况下配置为使用简单模型，但该模型会阻止备份事务日志。）
 - 使用以下排序规则设置创建数据库：Latin1_General_100_CI_AS_KS（其中 Latin1_General 因国家/地区而异，例如 Japanese_100_CI_AS_KS）。如果在数据库创建期间未指定此排序规则设置，则数据库内后续的服务架构创建操作将会失败，并显示类似于“<服务>: 架构需要不区分大小写的数据库”这样的错误（其中 <服务> 是要创建架构的服务的名称）。
 - 按照 [CTX137161](#) 中的说明启用读取已提交的快照。重要的是，该快照应在镜像数据库前启用，以避免错误。
3. 在 SQL Server A 上，将数据库备份到文件并将该文件复制到 SQL Server B。
4. 在 SQL Server B 上，将备份文件还原到该服务器 (SQL Server B) 中。
5. 在 SQL Server A 上，开始镜像。

下一步取决于 Citrix 管理员（即运行站点创建向导的人员）是否还具有完全数据库权限：

- 如果 Citrix 管理员有数据库权限（即数据库管理员和 Citrix 管理员是同一个人），Studio 将为您完成所有操作：
 1. Citrix 管理员使用 Studio 创建站点，指定之前创建的 SQL Server A 数据库的地址及其名称 (myDatabaseMirrorForXD)。
 2. 将自动应用数据库脚本，并设置主体数据库和镜像数据库。
- 如果 Citrix 管理员没有数据库权限，则必须从数据库管理员处获取帮助：
 1. Citrix 管理员使用 Studio 创建站点，指定之前创建的 SQL Server 数据库的地址及其名称 (myDatabaseMirrorForXD)。
 2. 在站点创建向导中，选择生成脚本将生成一个镜像脚本和一个主脚本。Citrix 管理员将这些脚本交给数据库管理员，数据库管理员将应用这些脚本（首先应用镜像脚本）。数据库管理员必须告诉 Citrix 管理员任务的完成时间。
 3. 返回 Studio，Citrix 管理员现在可以完成“创建站点”向导。此时主体数据库和镜像数据库已设置完毕。

要在创建站点后验证镜像，请运行 PowerShell cmdlet `get-configdbconnection`，以确保已在连接字符串中将故障转移合作伙伴设置为镜像。

如果您稍后在镜像的数据库环境中添加、移动或删除 Delivery Controller，请参阅[添加、删除或移动 Controller 或者移动 VDA](#)以了解注意事项。

如果站点中的所有 Delivery Controller 均出现故障，可以将 Virtual Delivery Agent 配置为在高可用性模式下运行，这样用户便可以继续访问并使用其桌面和应用程序。在高可用性模式下，VDA 将接受来自用户的直接 ICA 连接，而不是由 Controller 代理的连接。

此功能仅在与所有 Controller 的通信都出现故障时使用，这种情况非常罕见；此功能不能代替其他高可用性解决方案。有关详细信息，请参阅 [CTX127564](#)。

通过连接租用功能，用户可以连接以及重新连接到其最近使用的应用程序和桌面，即使在站点数据库不可用时也能连接，补充了 SQL Server 高可用性最佳实践。有关详细信息，请参阅[连接租用](#)。

交付方法

May 04, 2016

采用一种虚拟化部署满足每个用户的需要面临很多挑战。XenApp 和 XenDesktop 允许管理员采用各种方法（有时称为 FlexCast 模式）定制用户体验。

此交付方法集合（其中的交付方法分别有其各自的优势和劣势）可在任何用例场景下提供最佳用户体验。

现在，触摸屏设备（如平板电脑和智能手机）在移动性方面已标准化。这些设备在运行通常需要采用全屏并且需要依靠单击鼠标右键输入才能实现全部功能的基于 Windows 的应用程序时会出现问题。

采用 Citrix Receiver 的 XenApp 提供了安全的解决方案，允许移动设备用户访问其基于 Windows 的应用程序中的全部功能，无需针对本地移动平台重写这些应用程序。

XenApp 发布的应用程序交付方法利用 HDX 移动技术解决了与使 Windows 应用程序具有移动性相关的问题。此方法允许重构 Windows 应用程序以实现触摸体验，同时维护多点触控手势、本机菜单控件、摄像头和 GPS 功能等诸多功能。多种触控功能可在 XenApp 和 XenDesktop 中使用，无需更改任何应用程序源代码即可激活。

这些功能包括：

- 可编辑的字段获得焦点后自动显示键盘
- 更大的选取控件取代了 Windows 组合框控件
- 多点触控手势，如收缩和放大
- 可感知惯性的滚动功能
- 触控板或直接光标导航

升级物理计算机是一项艰巨的任何，很多企业每三到五年就要升级一次，尤其是当企业需要维护最新的操作系统和应用程序时。不断增长的企业还面临着向其网络中添加新计算机所产生的巨大成本开销。

VDI Personal vDisk 交付方法向任何使用服务器资源的计算机或瘦客户端上的单个用户提供完全个性化的桌面操作系统。管理员可以创建将其资源（如处理能力、内存和存储）存储在网络数据中心内的虚拟机。

这样可以延长旧计算机的使用时间，保持软件处于最新状态，以及在升级期间尽可能降低停机时间。

网络安全问题日益严重，在与需要访问公司应用程序和数据的承包商、合作伙伴和其他第三方临时工作人员合作时此情况尤为突出。工作人员可能还需要租用便携式计算机或其他设备，这可能会带来额外的成本忧虑。

数据、应用程序和桌面存储在采用 XenDesktop 和 XenApp 的安全网络的防火墙后面，这样一来，最终用户过渡时只需考虑用户设备输入和输出，如键盘、鼠标点击、音频和屏幕更新。通过在数据中心内维护这些资源，XenDesktop 和 XenApp 提供了比使用典型 SSL VPN 更加安全的远程访问解决方案。

通过具有个人虚拟磁盘的 VDI 部署，管理员可以在网络服务器上创建虚拟机并提供单用户桌面操作系统，从而利用瘦客户端或用户的个人设备。这使得 IT 部门无需购买昂贵的设备，即可维护与第三方工作人员合作时的安全性。

切换到新操作系统时，IT 可能会面临交付旧版和不兼容应用程序的挑战。

利用虚拟机托管应用程序，用户可以通过 Citrix Receiver 在升级后的虚拟机上运行旧的应用程序，并且不会出现任何兼容问题。这样一来，IT 无需花费额外的时间来解决和测试应用程序兼容问题，便于用户过渡，并使技术支持呼叫更加有效。

迁移期间使用 XenDesktop 的其他优势包括：

- 降低桌面复杂度
- 改善 IT 控制
- 在设备使用和工作区位置方面增强最终用户的灵活性

许多设计公司和制造公司严重依赖专业的 3-D 图形应用程序。要支持此类型的软件需要功能强大的硬件，由此产生的成本以及通过 FTP、电子邮件和类似的专门方法共享大型设计所带来的传输问题使得这些公司面临着财政压力。

XenDesktop 的托管物理桌面交付方法提供工作站和刀片式服务器的单个桌面映像，无需使用虚拟机管理程序在本地操作系统上运行图像密集型 3-D 应用程序。

所有文件存储在网络中的中央数据中心内，因此，在网络中向其他用户共享大型设计文件更快，更安全，因为无需再将文件从一个工作站传输到另一个工作站。

维护足够的员工以应对高峰期，同时又不会在不忙时过度配置计算机，这种需求为需要大规模呼叫中心的企业带来了严峻挑战。

池 VDI 交付方法可在置备大量用户的情况下，以最低成本动态地向多个用户提供标准桌面的访问权限。池计算机按照每个会话以先到先服务的原则进行分配。

由于在用户注销时会丢弃执行的所有更改，因此，基本上无需每日对这些虚拟机进行管理。这同样也增强了安全性。

XenApp 托管桌面交付方法是另一种转换呼叫中心的可行选项。此方法在单个基于服务器的操作系统上托管多个用户桌面。

此方法比池 VDI 更具成本效益，但是使用 XenApp 托管桌面时，会限制用户安装应用程序、更改系统设置和重新启动服务器。

新部署

Nov 02, 2016

To build a XenApp or XenDesktop deployment:

1. Set up the virtualization environment to host and manage the components of your XenApp or XenDesktop environment. See [System requirements](#) for supported versions of the virtualization platforms, management tools, and cloud deployment solutions listed here.

You can use these virtualization platforms to host and manage machines in your XenApp or XenDesktop environment:

- XenServer. See [XenServer](#) for information on setting up and using XenServer.
- VMware vSphere. See [Prepare the virtualization environment: VMware](#) for guidance on setting up and using VMware vSphere with XenApp or XenDesktop.
- Hyper-V with Microsoft System Center Virtualization Machine Manager (VMM). See [Prepare the virtualization environment: Microsoft System Center Virtual Machine Manager](#) for guidance on setting up and using Hyper-V with VMM with XenApp or XenDesktop.

You can use Microsoft System Center Configuration Manager with Citrix Connector 7.5 for System Center Configuration Manager 2012 to manage physical and virtual machines in your XenApp or XenDesktop environment or use it to enable the Wake on LAN feature of Remote PC Access. See [Prepare for using Microsoft System Center Configuration Manager](#).

You can use these cloud deployment solutions to host product components and provision virtual machines. These solutions pool computing resources to build public, private, and hybrid Infrastructure as a Service (IaaS) clouds.

- Amazon Web Services, see [Deploy XenApp and XenDesktop 7.5 and 7.6 with Amazon VPC](#).
- Citrix CloudPlatform, see [XenApp and XenDesktop concepts and deployment on CloudPlatform](#).

2. Set up the non-Citrix infrastructure components required to build your XenApp or XenDesktop Site. These include at least one domain controller running Active Directory Domain Services.
3. Install the Citrix components that make up your XenApp or XenDesktop Site. You can install components using a wizard-based graphical interface or a command-line interface, which enables scripted installation. Both methods install most prerequisites automatically.
 1. Before beginning any installation, review the [System requirements](#). Also, read and complete the [Prepare to install](#) checklist.
 2. Install the core components: Delivery Controller, Citrix Studio, Citrix Director, Citrix License Server, and Citrix StoreFront. See [Install using the graphical interface](#) or [Install using the command line](#) for information on installing these components.
 3. From Studio, create a Site. See [Create a Site](#).
 4. Install a Virtual Delivery Agent (VDA), either on the master image you will use to create virtual machines or directly on each machine. See [Install using the graphical interface](#) or [Install using the command line](#) for information on installing the VDA. You may also want to see [Install or remove Virtual Delivery Agents using scripts](#).
For Remote PC Access deployments, install a VDA for Desktop OS on each office PC. Citrix recommends using the VDA installer's command line interface and your existing Electronic Software Distribution (ESD) methods.
 5. Optionally, install the Universal Print Server on the print servers in your environment. See [Install using the graphical interface](#) or [Install using the command line](#) for information on installing the Universal Print Server.
4. Optionally, integrate additional Citrix components into your XenApp or XenDesktop deployment. For example:

- Provisioning Services is an optional component of XenApp and XenDesktop that provisions machines by streaming a master image to target devices. See [Provisioning Services](#).
 - Citrix NetScaler Gateway is a secure application access solution that provides administrators granular application-level policy and action controls to secure access to applications and data. See [Citrix NetScaler Gateway](#).
 - Citrix CloudBridge is a set of appliances that optimize WAN performance. See [Citrix CloudBridge](#).
5. Set up the resources you will deliver to users. How you do this depends on the delivery method you are using, but this is the basic sequence for most delivery methods:
1. Using your hypervisor's management tool, create a master image that defines the desktops or applications you want to provide. See [Prepare a master image](#).
 2. Create a machine catalog containing physical and virtual machines from that master image. See [Create a machine catalog](#).
 - If you are using Machine Creation Services to provision machines, you can add machines to the machine catalog from within Studio.
 - If you are using Provisioning Services to provision machines, you add machines to the machine catalog from the Provisioning Services console.
 3. From Studio, create a Delivery Group to specify which users can access these machines and the applications installed on them. See [Delivery groups](#).

准备安装

Jul 13, 2016

下表列出了安装核心组件（Delivery Controller、Citrix Studio、Citrix Director、Citrix 许可证服务器、StoreFront）和 Virtual Delivery Agent (VDA) 之前需要完成的任务以及需要注意的事项。

• 说明	
首先： • 如果对产品不熟悉，请阅读技术预览和相关内容。 • 查阅 — 已知问题 以了解可能会遇到的安装问题。 • 如果要在云环境中安装组件，请参阅： • Deploy XenApp and XenDesktop 7.5 and 7.6 with Amazon VPC（使用 Amazon VPC 部署 XenApp 和 XenDesktop 7.5 和 7.6），面向 Amazon Web Services； • XenApp and XenDesktop concepts and deployment on CloudPlatform（XenApp 和 XenDesktop 概念和在 CloudPlatform 上的部署），面向 Citrix CloudPlatform。 • 如果要将 XenServer 用于您的虚拟环境，请参阅 XenServer 文档以获取相关指南。 • 如果要将 VMware 或 Microsoft System Center Virtual Machine Manager 用于您的虚拟环境，请参阅链接的文档。	
确定组件的安装位置，然后准备计算机和操作系统。 • 参阅系统要求以了解 Controller、Studio、Director、虚拟化资源和 VDA 支持的操作系统及版本。Citrix StoreFront 和 Citrix 许可证服务器要求文档指定了支持的平台。 • 您可以将核心组件安装在同一服务器或不同服务器上。例如，要远程管理较小的部署，可以将 Studio 与 Controller 安装在不同服务器上。为适应将来的扩展，请考虑将组件安装在单独的服务器上；例如，将许可证服务器和 Director 安装在不同的服务器上。 • 可以在相同的服务器上安装 Delivery Controller 和适用于 Windows 服务器操作系统的 Virtual Delivery Agent。启动安装程序并选择 Delivery Controller（以及要在该计算机上安装的任何其他核心组件），然后再次启动安装程序并选择适用于 Windows 服务器操作系统的 Virtual Delivery Agent。 • 请勿在运行 XenApp 6.5 Feature Pack 2 for Windows Server 2008 R2 或任何早期版本的 XenApp 的服务器上安装 Studio。 • 确保每个操作系统都具有最新更新。 • 确保所有计算机具有同步的系统时钟。对于 Kerberos 基础结构而言，必需实现同步，以保护计算机之间的通信。 • 组件默认安装在 C:\Program Files\Citrix 中。可以在安装过程中指定其他位置，但该位置必须具有网络服务的执行权限。 • 大多数必备组件会自动安装；但是， — 系统要求 文档中指出了例外情况。	
确定站点配置数据库的 SQL Server 软件的安装位置。 • 默认情况下，在您安装 Controller 时，如果未检测到另一个实例，SQL Server 2012 Express 将自动安装在服务器上。 默认安装程序使用默认 Windows 服务帐户和权限。请参阅 Microsoft 文档了解关于这些默认设置的详细信息，其中包括如何向 sysadmin 角色添加 Windows 服务帐户。Controller 使用此配置中的网络服务帐户。Controller 不需要使用任何其他 SQL Server 角色或权限。 如果需要，您可以为数据库实例选择隐藏实例。在 Studio 中配置数据库的地址时，请输入实例的静态端口号，而不是它的名称。请参阅 Microsoft 文档了解关于隐藏 SQL Server 数据库引擎实例的详细信息。 • 或者，可以在该服务器或其他服务器上单独安装支持的 SQL Server 版本。在此类情况下，不需要在安装核心组件之前安装 SQL Server 软件，但必须在创建站点之前安装。 • 查阅 — 规划 文档中的数据库注意事项，并设置任意受支持的冗余基础结构。 重要：需要在 Controller 与数据库之间执行 Windows 身份验证。	

• 说明	确定端口的打开方式。
	默认情况下，如果 Windows 防火墙服务正在运行，那么即使未启用防火墙，以下端口也会自动打开。如果使用第三方防火墙或未使用防火墙，或者只是因为更喜欢亲自打开，则可以禁用此默认操作，然后自动打开端口。 • Controller : TCP 80、443 • Director : TCP 80、443 • 许可证服务器 : TCP 7279、8082、8083、27000 • StoreFront : TCP 80、443 提示：有关完整的端口信息，请参阅CTX101810。有关其他安装选项，请参阅使用命令行安装。
	配置 Active Directory 域。 • 除了是域用户以外，您还必须是正在安装核心组件的计算机上的本地管理员。 • 请勿尝试在域控制器上安装任何组件。 • — 系统要求 文档列出了支持的功能级别。请参阅 Microsoft 文档以了解相关说明。 安装许可证服务器时，该用户帐户会自动成为许可证服务器上的完全权限管理员。
	安装 Director 之前，确定是否使用 Director 的重影功能，该功能使用 Windows 远程协助。
	须知事项： • 如果组件未成功安装，此过程将停止并显示一条错误消息。成功安装的组件将会保留，不需要重新安装。 • Studio 在安装后自动启动。可以在安装过程中禁用此操作。 • 在安装前、安装期间和安装完毕后创建对象时，最好为每个对象（例如，网络、组、目录、资源）指定唯一的名称。 • 在 Amazon Web Services (AWS) 中安装组件后，在使用 Studio 创建站点时，需要知道地理区域、可用区域、VPC 名称、子网地址、域名、安全组名称及凭据。

• 说明	
	如果要安装 VDA for Windows Desktop OS，请确定是否要安装 HDX 3D Pro 版。 HDX 3D Pro 功能交付可通过 GPU 实现最佳硬件加速的桌面和应用程序。有关详细信息，请参阅 HDX 3D Pro 文档。
	确定 VDA 的使用方式。 默认设置假定您将已安装 VDA 的主映像与 Machine Creation Services 或 Provisioning Services 结合使用，以创建其他虚拟机。如果要在现有计算机上安装 VDA，可以覆盖此默认设置。
	确定是否要安装 Citrix Receiver for Windows (CitrixReceiver.exe)。 可以禁用此默认操作。
	确定端口的打开方式。 默认情况下，如果 Windows 防火墙服务正在运行，那么即使未启用防火墙，以下端口也会自动打开。如果使用第三方防火墙或未使用防火墙，或者只是因为更喜欢亲自打开，则可以禁用此默认操作，然后自动打开端口。 • Controller : TCP 80、1494、2598、8008 • 要在用户设备与虚拟桌面之间进行通信，请将端口 1494 和 2598 上的入站 TCP 配置为端口例外。为安全起

	说明 见, Citrix 建议您不要将这些已注册的端口用于除 ICA 协议和通用网关协议之外的任何用途。 ● 要在 Controller 与虚拟桌面之间进行通信, 请将入站端口 80 配置为端口例外。
	● Windows 远程协助: TCP 3389 如果此功能已启用, Windows 会自动打开此端口, 即使您选择手动打开端口也是如此。 ● 实时音频传输: UDP 16500-16509 提示: 有关端口的完整信息, 请参阅 CTX101810。
	确定如何指定已安装 Controller 的位置。 ● 手动, 通过输入 Controller 的完全限定的域名 (FQDN)。虽然可以指定当前不在域中的 Controller, 但 Controller 必须在域中才能与 VDA 连接。此外, 只能对域中的 Controller 执行连接测试。 ● 使用 Active Directory (如果 Controller 在域中)。 ● 允许 Machine Creation Services 指定 Controller。 ● 稍后, 通过重新运行安装程序、使用 Citrix 策略、设置注册表值或使用 Active Directory OU。 用于指定 Controller 位置的 Citrix 组策略设置会覆盖在安装期间提供的设置。 在初次指定 Controller 位置后, 可以在安装其他 Controller 后使用自动更新功能更新 VDA。
	确定是否要使用以下功能: ● 优化性能: 此功能处于启用状态时, 将对虚拟机管理程序上的 VM 中运行的 VDA 使用优化工具。VM 优化包括禁用脱机文件、禁用后台碎片整理, 以及降低事件日志大小。有关详细信息, 请参阅 CTX125874。如果要使用 Remote PC Access, 请勿启用此选项。默认情况下已启用。 ● Windows 远程协助: 此功能处于启用状态时, Windows 远程协助将与 Director 的用户重影功能结合使用, 并且 Windows 会自动打开防火墙中的 TCP 端口 3389, 即使您选择手动打开防火墙端口也是如此。默认情况下已启用。 ● 对音频使用实时音频传输: 此功能处于启用状态时, 将对音频数据包使用 UDP, 这样可以提升音频性能。默认情况下已启用。 ● Personal vDisk: (仅在 VM 上安装 VDA for Windows Desktop OS 时可用)。此功能处于启用状态时, 个人虚拟磁盘可以与主映像结合使用。有关详细信息, 请参阅 Personal vDisks。默认情况下已禁用。
	须知事项: ● 默认情况下在 Windows Server 上启用后台打印程序服务。如果禁用此服务, 将无法成功安装 VDA for Windows Server OS。因此, 安装 VDA 前, 应确保启用此服务。 ● 安装程序会自动检测您的操作系统, 并允许您仅在该系统上安装支持的 VDA 类型: VDA for Windows Server OS 或 VDA for Windows Desktop OS。 ● VDA 安装期间会同时安装 Profile Management。 ● 安装 VDA 时, 系统将自动创建名为“直接访问用户”的新本地用户组。在 VDA for Windows Desktop OS 上, 此用户组仅适用于 RDP 连接; 在 VDA for Windows Server OS 上, 此用户组则适用于 ICA 和 RDP 连接。 ● 安装 VDA for Windows Server OS 时, Remote Desktop Services 角色服务会自动安装并启用 (如果尚未安装和启用)。 ● 对于 Remote PC Access 配置, 请在用户将远程访问的每台物理办公室 PC 上安装 VDA for Windows Desktop OS。 ● 作为使用完整产品 ISO 安装 VDA 的替代方法, 您可以使用独立的 VDA 安装软件包。有关详细信息, 请参阅使用独立的软件包安装 VDA。

Windows XP 或 Windows Vista 系统不支持最新的 Virtual Delivery Agent (VDA)。此外，本版本（和其他最近版本）中的某些功能无法在这些操作系统中使用。要使用此版本中的所有功能，Citrix 建议您将 Windows XP 或 Windows Vista 系统替换为 Windows 7、Windows 8 或 Windows 10，然后安装此版本中的 Virtual Delivery Agent。

如果您必须继续容纳运行 Windows XP 或 Windows Vista 的计算机，则为了满足这些要求，可以安装早期 Virtual Desktop Agent 版本（具有某些修补程序的 5.6 FP1）。有关详细信息，请参阅 [CTX140941](#)。

请谨记：

- 不能在 Windows XP 或 Windows Vista 系统上安装核心组件（Controller、Studio、Director、StoreFront、Citrix 许可证服务器）。
- Windows Vista 系统不支持 Remote PC Access。
- Microsoft 已于 2014 年 4 月 8 日终止对 Windows XP 提供支持，此时 Citrix 不再支持 Windows XP。
- 继续使用旧的 VDA 可能会影响功能的可用性以及向 Controller 注册 VDA，请参阅[混合环境注意事项](#)。

准备虚拟化环境：VMware

Nov 02, 2016

Follow this guidance if you use VMware to provide virtual machines.

1. Install vCenter Server and the appropriate management tools. (No support is provided for vSphere vCenter Linked Mode operation.)
2. Create a VMware user account with the following permissions, at the DataCenter level, at a minimum. This account has permissions to create new VMs and is used to communicate with vCenter.

SDK	User Interface
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
System.Anonymous, System.Read, and System.View	Added automatically.
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 SDK	Virtual machine > Configuration > Remove disk User interface
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 virtual machine
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 and vSphere 5.1, Update 1: Virtual machine > State > Create snapshot vSphere 5.5: Virtual machine > Snapshot management > Create snapshot

SDK VirtualMachine.State.RemoveSnapshot	User Interface vSphere 5.0, Update 2 and 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 and vSphere 5.1, Update 1: Virtual machine > State > Revert to snapshot
	vSphere 5.5: Virtual machine > Snapshot management > Revert to snapshot

3. If you want the VMs you create to be tagged, add the following permissions for the user account:

SDK	User Interface
Global.ManageCustomFields	Global > Manage custom attributes
Global.SetCustomField	Global > Set custom attribute

To ensure that you use a clean base image for creating new VMs, tag VMs created with Machine Creation Services to exclude them from the list of VMs available to use as base images.

To protect vSphere communications, Citrix recommends that you use HTTPS rather than HTTP. HTTPS requires digital certificates. Citrix recommends you use a digital certificate issued from a certificate authority in accordance with your organization's security policy.

If you are unable to use a digital certificate issued from a certificate authority, and your organization's security policy permits it, you can use the VMware-installed self-signed certificate. Add the VMware vCenter certificate to each Controller. Follow this procedure:

1. Add the fully qualified domain name (FQDN) of the computer running vCenter Server to the hosts file on that server, located at %SystemRoot%/WINDOWS/system32/Drivers/etc/. This step is required only if the FQDN of the computer running vCenter Server is not already present in the domain name system.
2. Obtain the vCenter certificate using any of the following methods:
 - From the vCenter server:
 1. Copy the file rui.crt from the vCenter server to a location accessible on your Delivery Controllers.
 2. On the Controller, navigate to the location of the exported certificate and open the rui.crt file.
 - Download the certificate using a web browser. If you are using Internet Explorer, depending on your user account, you may need to right-click on Internet Explorer and choose Run as Administrator to download or install the certificate.
 1. Open your web browser and make a secure web connection to the vCenter server; for example <https://server1.domain1.com>
 2. Accept the security warnings.
 3. Click on the address bar where it shows the certificate error.

4. View the certificate and click on the Details tab.
 5. Select Copy to file and export in .CER format, providing a name when prompted to do so.
 6. Save the exported certificate.
 7. Navigate to the location of the exported certificate and open the .CER file.
- Import directly from Internet Explorer running as an administrator:
 1. Open your web browser and make a secure web connection to the vCenter server; for example <https://server1.domain1.com>.
 2. Accept the security warnings.
 3. Click on the address bar where it shows the certificate error.
 4. View the certificate.
 - Import the certificate into the certificate store on each of your Controllers:
 1. Click Install certificate, select Local Machine, and then click Next.
 2. Select Place all certificates in the following store, and then click Browse.
 3. If you are using Windows Server 2008 R2:
 1. Select the Show physical stores check box.
 2. Expand Trusted People.
 3. Select Local Computer.
 4. Click Next, then click Finish.
 - If you are using Windows Server 2012 or Windows Server 2012 R2:
 1. Select Trusted People, then click OK.
 2. Click Next, then click Finish.

Important: If you change the name of the vSphere server after installation, you must generate a new self-signed certificate on that server before importing the new certificate.

Use a master VM to provide user desktops and applications.

1. Install a VDA on the master VM, selecting the option to optimize the desktop, which improves performance.
2. Take a snapshot of the master VM to use as a back-up. For more information, see [Prepare a master image](#).

If you are using Studio to create VMs, rather than selecting an existing machine catalog, specify the following information when setting up your hosting infrastructure to create virtual desktops.

1. Select the VMware vSphere host type.
2. Enter the address of the access point for the vCenter SDK (<https://vmware.example.com/sdk>).
3. Enter the credentials for the VMware user account you set up earlier that has permissions to create new VMs. Specify the username in the form domain/username.

准备虚拟化环境：Microsoft System Center Virtual Machine Manager

May 04, 2016

如果您结合使用 Hyper-V 与 Microsoft System Center Virtual Machine Manager (VMM) 来提供虚拟机，请按本指导操作。

此版本支持：

- VMM 2012 — 提供改进的管理功能，使您能够管理整个虚拟化数据中心以及虚拟机。此版本现在可安排群集主机修补以及与 Windows Server Update Services 的集成，以允许您定义每个主机所需的修补程序的基线。
- VMM 2012 SP1 — 在具有群集共享卷和存储区域网络 (SAN) 的文件服务器上使用 SMB 3.0 时，可提供有关 Machine Creation Services (MCS) 的性能改进。这些文件共享可在 SAN 存储上提供低成本缓存并降低 IO，从而提高性能。
- VMM 2012 R2 — 实现了主要 Windows Server 2012 R2 功能的现有规模管理，包括运行 VM 快照、动态调整 VHDX 的大小以及存储空间。

此版本仅支持安装了 VMM 2012 R2 的第一代虚拟机。Machine Creation Services (MCS) 和 Provisioning Services 部署不支持第二代虚拟机。使用 MCS 或 Provisioning Services 创建 VM 时，第二代 VM 不出现在主 VM 的选择列表中；它们默认启用 Secure Boot，从而防止 VDA 正常工作。

- 从 VMM 2012 升级至 VMM 2012 SP1 或 VMM 2012 R2
有关 VMM 和 Hyper-V 主机的要求，请参阅 <http://technet.microsoft.com/zh-cn/library/gg610649.aspx>。有关 VMM 控制台的要求，请参阅 <http://technet.microsoft.com/zh-cn/library/gg610640.aspx>。

不支持混合 Hyper-V 群集。混合群集的一个示例：群集一部分运行 Hyper-V 2008，而另一部分则运行 Hyper-V 2012。

- 从 VMM 2008 R2 升级到 VMM 2012 SP1
如果从 VMM 2008 R2 上的 XenDesktop 5.6 进行升级，请遵循以下顺序以避免 XenDesktop 停机。
 1. 将 VMM 升级到 2012（现在运行的是 XenDesktop 5.6 和 VMM 2012）
 2. 将 XenDesktop 升级到最新版本（现在运行的是最新 XenDesktop 和 VMM 2012）
 3. 将 VMM 从 2012 升级到 2012 SP1（现在运行的是最新的 XenDesktop 和 VMM 2012 SP1）
- 从 VMM 2012 SP1 升级到 VMM 2012 R2
如果从 VMM 2012 SP1 上的 XenDesktop 或 XenApp 7.x 开始升级，请遵循以下顺序以避免 XenDesktop 停机。
 1. 将 XenDesktop 或 XenApp 升级到最新版本（现在运行的是最新 XenDesktop 或 XenApp 和 VMM 2012 SP1）
 2. 将 VMM 2012 SP1 升级到 2012 R2（现在运行的是最新 XenDesktop 或 XenApp 和 VMM 2012 R2）

1. 安装和配置虚拟机管理程序。

1. 在服务器上安装 Microsoft Hyper-V Server 和 VMM。所有 Delivery Controller 必须与 VMM 服务器位于同一个林中。
2. 在所有 Controller 上安装 System Center Virtual Machine Manager 控制台。
3. 验证以下帐户信息：
 - 用于在 Studio 中指定主机的帐户是相关 Hyper-V 计算机的 VMM 管理员或 VMM 委派管理员。如果此帐户在 VMM 中仅具有委派管理员角色，则在主机创建过程中不会在 Studio 中列出存储数据。
 - 用于 Studio 集成的用户帐户还必须属于每个 Hyper-V Server 上的管理员本地安全组的成员，才能支持 VM 生命周期管理（例如 VM 创建、更新和删除）。

注意：不支持在运行 Hyper-V 的服务器上安装 Controller。

2. 创建主 VM。

1. 在主 VM 上安装 Virtual Delivery Agent，然后选择用于优化桌面的选项。这样会提高性能。
 2. 生成主 VM 的快照作为备份。
- 有关详细信息，请参阅[准备主映像](#)。

3. 创建虚拟桌面。如果要使用 MCS 创建 VM，创建站点或连接时，

1. 选择 Microsoft 虚拟化主机类型。
2. 以主机服务器的完全限定的域名形式输入地址。
3. 输入先前设置的管理员帐户的凭据，该帐户应具有创建新 VM 的权限。
4. 在主机详细信息对话框中，选择创建新 VM 时将使用的群集或独立主机。

重要：即使使用单 Hyper-V 主机部署，也请浏览并选择群集或独立主机。

对于使用 VM 存储的 SMB 3 文件共享上的 MCS 创建的计算机目录，请确保按照以下要求设置凭据，以便来自 Controller 的虚拟机管理程序通信库 (HCL) 的调用能够成功连接到 SMB 存储：

- VMM 用户凭据必须包含对 SMB 存储的完全读取写入权限。
- VM 存储生命周期事件期间的存储虚拟磁盘操作通过 Hyper-V 服务器使用 VMM 用户凭据执行。

如果将 SMB 用作存储，请在 Windows Server 2012 上同时使用 VMM 2012 SP1 和 Hyper-V 时启用 Controller 到单个 Hyper-V 计算机的身份验证凭据安全支持提供程序 (CredSSP)。有关详细信息，请参阅 [CTX137465](#)。

使用标准 PowerShell V3 远程会话，HCL 可使用 CredSSP 打开与 Hyper-V 计算机的连接。此功能可将 Kerberos 加密的用户凭据传递到 Hyper-V 计算机，并在使用所提供的凭据（本例中指 VMM 用户的凭据）运行的远程 Hyper-V 计算机上的此会话中传递 PowerShell 命令，以便存储的通信命令正确运行。

以下任务使用的 PowerShell 脚本源于 HCL，随后将被发送到 Hyper-V 计算机以作用于 SMB 3.0 存储。

- **合并主映像** - 主映像可创建新的 MCS 置备方案（计算机目录）。它将克隆并展平主 VM，以便准备好从新创建的磁盘创建新 VM（并删除对初始主 VM 的依赖）。

ConvertVirtualHardDisk 位于 root\virtualization\v2 命名空间

示例：

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

- **创建差异磁盘** - 从合并主映像时产生的主映像创建差异磁盘。差异磁盘随后将连接到新 VM。

CreateVirtualHardDisk 位于 root\virtualization\v2 命名空间

示例：

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

- **上载身份磁盘** - HCL 不能直接将身份磁盘上载到 SMB 存储。因此，Hyper-V 计算机必须将身份磁盘上载并复制到该存储。由于 Hyper-V 计算机无法从 Controller 中读取磁盘，因此 HCL 必须首先通过 Hyper-V 计算机复制身份磁盘，如下所述。
 1. HCL 通过管理员共享将身份上载到 Hyper-V 计算机。
 2. Hyper-V 计算机通过 PowerShell 远程会话中运行的 PowerShell 脚本将磁盘复制到 SMB 存储。将在 Hyper-V 计算机上创建一个文件夹，此文件夹的权限已锁定，仅 VMM 用户有权访问（通过远程 PowerShell 连接）。
 3. HCL 删除管理员共享中的文件。
 4. HCL 完成身份磁盘到 Hyper-V 计算机的上载后，远程 PowerShell 会话可将身份磁盘复制到 SMB 存储，然后将其从 Hyper-V 计算机中删除。如果删除了身份磁盘文件夹，将重新创建以便重复使用。
- **下载身份磁盘** - 与上载一样，身份磁盘通过 Hyper-V 计算机传递到 HCL。以下过程将在 Hyper-V 服务器上创建一个仅 VMM 用户有权访问的文件夹（如果尚不存在）。
 1. Hyper-V 计算机通过 PowerShell V3 远程会话中运行的 PowerShell 脚本将磁盘从 SMB 存储复制到本地 Hyper-V 存储。
 2. HCL 将 Hyper-V 计算机管理员共享中的磁盘读取到内存。
 3. HCL 删除管理员共享中的文件。
- **创建个人虚拟磁盘** - 如果管理员在个人虚拟磁盘计算机目录中创建 VM，则必须创建一个空磁盘 (PvD)。

创建空磁盘的调用无需直接访问存储。如果您所具有的 PvD 磁盘与主磁盘或操作系统磁盘位于不同的存储，请使用远程 PowerShell 在与创建的 VM 具有相同名称的目录文件夹中创建 PvD 磁盘。对于 CSV 或 LocalStorage，请勿使用远程 PowerShell。在创建空磁盘之前创建该目录可避免 VMM 命令失败。

对于 Hyper-V 计算机，请在存储上执行 mkdir。

准备使用 Microsoft System Center Configuration Manager

May 04, 2016

通过这些集成选项，使用 System Center Configuration Manager (Configuration Manager) 2012 来管理对物理设备上的应用程序和桌面的访问的站点可以将这种用法扩展到 XenApp 或 XenDesktop。

- **Citrix Connector 7.5 for Configuration Manager 2012** – Citrix Connector 在 Configuration Manager 与 XenApp 或 XenDesktop 之间提供桥接。利用 Connector，您可以在 Configuration Manager 管理的各个物理环境与 XenApp 或 XenDesktop 管理的各个虚拟环境之间统一执行日常操作。有关 Connector 的信息，请参阅 [Citrix Connector 7.5 for System Center Configuration Manager 2012](#)。
- **Configuration Manager 唤醒代理功能** – 无论您的环境是否包含 Citrix Connector，使用 Remote PC Access 局域网唤醒功能均需要 Configuration Manager。有关详细信息，请参阅 [Configuration Manager 与 Remote PC Access 局域网唤醒](#)。
- **XenApp 和 XenDesktop 属性** – 利用 XenApp 和 XenDesktop 属性，您可以识别 Citrix 虚拟桌面以便通过 Configuration Manager 进行管理。Citrix Connector 会自动使用这些属性，但是，您也可以进行手动配置，如以下部分所述。

属性可供 Microsoft System Center Configuration Manager 2012 和 2012 R2 管理虚拟桌面之用。

在 Configuration Manager 2012 中显示的布尔属性可能会显示为 1 或 0，而非 true 或 false。

这些属性适用于 Root\Citrix\DesktopInformation 命名空间中的 Citrix_virtualDesktopInfo 类。属性名称来源于 Windows Management Instrumentation (WMI) 提供程序。

属性	说明
AssignmentType	设置 IsAssigned 的值。有效值为： ● ClientIP ● ClientName ● 无 ● “用户”– 将 IsAssigned 设置为 True
BrokerSiteName	站点；返回的值与 HostIdentifier 相同。
DesktopCatalogName	与桌面关联的计算机目录。
DesktopGroupName	与桌面关联的交付组。
HostIdentifier	站点；返回的值与 BrokerSiteName 相同。
IsAssigned	如果为 True，则将桌面分配给用户，对于随机桌面则设置为 False。
IsMasterImage	允许有关环境的决定。例如，您可能希望在主映像而非置备的计算机上安装应

属性	说明
	用程序，尤其是当这些计算机在引导计算机上处于干净状态时。有效值为： • True：在用作主映像的 VM 上（该值在安装期间基于选项而设置）。 • Cleared：在从该映像置备的 VM 上。
IsVirtualMachine	如果是虚拟机，则为 True；如果是物理机，则为 False。
OSChangesPersist	如果桌面操作系统映像在每次重新启动时都重置为清除状态，则为 False，否则为 True。
PersistentDataLocation	Configuration Manager 存储永久数据的位置。用户无法访问此位置。
PersonalVDiskDriveLetter	对于具有个人虚拟磁盘的桌面，是指分配给个人虚拟磁盘的驱动器盘符。
BrokerSiteName、 DesktopCatalogName、 DesktopGroupName、HostIdentifier	在桌面向 Controller 注册时确定；对于未完全注册的桌面，这些属性为空。

要收集属性，请在 Configuration Manager 运行硬件清单。要查看属性，请使用 Configuration Manager 资源浏览器。在这些实例中，名称可包括空格或与属性名称略不相同。例如，**BrokerSiteName** 可能会显示为 Broker Site Name。有关以下任务的信息，请参阅 [Citrix WMI 属性](#) 和 [System Center Configuration Manager 2012](#)：

- 配置 Configuration Manager 以便从 Citrix VDA 收集 Citrix WMI 属性
- 使用 Citrix WMI 属性创建基于查询的设备集合
- 根据 Citrix WMI 属性创建全局条件
- 使用全局条件定义应用程序部署类型要求

还可以使用 Root\ccm_vdi 命名空间中 Microsoft 类 CCM_DesktopMachine 中的 Microsoft 属性。有关详细信息，请参阅 Microsoft 文档。

有关规划和交付 Remote PC Access 的信息，请参阅 [Remote PC Access](#) 和 [向用户提供 Remote PC Access](#)。

要配置 Remote PC Access 局域网唤醒功能，在办公室 PC 机上安装 VDA 并使用 Studio 创建或更新 Remote PC Access 部署前，请完成以下操作：

- 在组织中配置 Configuration Manager 2012，然后将 Configuration Manager 客户端部署到所有 Remote PC Access 计算机，从而使所安排的 SCCM 清单周期有时间运行（或在需要时强制运行一个周期）。在 Studio 中指定用来配置 Configuration Manager 连接的访问凭据必须包含该作用域中的集合以及 Remote Tools Operator 角色。
- 要支持 Intel 主动管理技术 (AMT)：
 - PC 上的最低受支持版本必须为 AMT 3.2.1。
 - 通过证书及相关置备进程置备 PC，以便使用 AMT。
- 要支持 Configuration Manager 唤醒代理和/或幻数据包功能：
 - 在每台 PC 的 BIOS 设置中配置局域网唤醒功能。
 - 要获取 Configuration Manager 唤醒代理支持，请在 Configuration Manager 中启用此选项。对于组织中将使用 Remote PC Access 局域网唤醒功能的 PC 所属的每个子网，请确保有三台或更多的计算机可以作为标记计算机使用。

- 要支持幻数据包功能，请将网络路由器和防火墙配置为允许使用子网定向的广播或单播发送幻数据包。

在办公室 PC 上安装 VDA 后，在 Studio 中创建 Remote PC Access 部署时，请启用或禁用电源管理。

- 如果启用电源管理，请指定详细的连接信息：Configuration Manager 地址和访问凭据以及名称。
- 如果不启用电源管理，可在以后添加电源管理 (Configuration Manager) 连接，然后编辑 Remote PC Access 计算机目录，以启用电源管理并指定新的电源管理连接。

可以编辑电源管理连接以配置使用 Configuration Manager 唤醒代理和幻数据包功能，以及更改数据包传输方式。

使用图形界面进行安装

May 04, 2016

开始执行任何安装之前，请阅读并完成[准备安装](#)中的任务。

启动安装程序图形界面：

1. 下载产品包并进行解压缩。或者刻录 ISO 文件的 DVD。
2. 通过本地管理员帐户，登录要在安装组件的服务器。
3. 在驱动器中插入 DVD 或装载 ISO 文件。如果安装程序未自动启动，请双击 AutoSelect 应用程序或已装载的驱动器。
4. 选择要安装的组件：
 - 如果刚刚开始安装，请选择 Delivery Controller。借助该组件，可以在同一台服务器上安装 Delivery Controller 以及 Studio、Director、许可证服务器和 StoreFront，后四项组件为可选项。
 - 如果已经安装了一些组件并且希望对部署进行扩展，请在右侧的列中单击要安装的组件。该列提供了核心组件以及通用打印服务器，后者可安装到打印服务器上。
 - 要安装 Virtual Delivery Agent (VDA)，请单击可用的 VDA 项，安装程序可以为运行安装程序的操作系统选择合适的版本。

之后，如果要自定义已安装的 VDA：

1. 从用于删除或更改程序的 Windows 功能，选择 Citrix Virtual Delivery Agent <版本号>，然后单击鼠标右键并选择更改。
2. 选择自定义 Virtual Delivery Agent 设置。安装程序启动后，可更改 Controller 地址、向注册 Controller 的 TCP/IP 端口（默认值 = 80）或者是否自动开启 Windows Firewall 端口例外。

也可以使用图形界面升级组件；请参阅[升级部署](#)。

作为使用完整产品 ISO 安装 VDA 的替代方法，您可以使用独立的 VDA 安装软件包。有关详细信息，请参阅[使用独立的软件包安装 VDA](#)。

在 XenApp 和 XenDesktop 7.6 FP3 中，UPS 软件包包含独立 UPS 客户端和服务器的更新版本。有关安装说明，请参阅[准备打印机](#)中的“安装 Citrix 通用打印服务器 (UPS)”部分。

使用命令行安装

Nov 02, 2016

Use the command line interface to:

- Install one or more core components: Delivery Controller, Citrix Studio, Citrix Director, License Server, and StoreFront.
- Install a Virtual Delivery Agent (VDA) on a master image or on a virtual or physical machine.
You can also customize scripts provided on the media, then use them to install and remove VDAs in Active Directory.
- Customize a previously-installed VDA.
- Install a Universal Print Server, which provisions network session printers. The Controller already has the Universal Print Server functionality; you need only install the Universal Print Server on the print servers in your environment.

You can also:

- Remove components from this version that you previously installed, using the /remove or /removeall options. For details, see [Remove components](#).
- Upgrade components; for details, see [Upgrade a deployment](#).

To see command execution progress and return values, you must be the original administrator or use 'Run as administrator.' For more information, see the Microsoft command documentation.

Important: Before beginning an installation, read and complete the tasks in [Prepare to install](#).

From the \x64\XenDesktop Setup directory on the media, run the XenDesktopServerSetup.exe command. The following table describes command options.

Note: To install XenApp, include the /xenapp option on the command line. To install XenDesktop, do not include the /xenapp option.

Option	Description
/help or /h	Displays command help.
/quiet or /passive	No user interface appears during the installation. The only evidence of the installation process is in Windows Task Manager. If this option is omitted, the graphical interface launches.
/logpath path	Log file location. The specified folder must already exist; the installer does not create it. Default = "%TEMP%\Citrix\XenDesktop Installer"
/noreboot	Prevents a restart after installation. (For most core components, a restart is not enabled by default.)
/remove	Removes the core components specified with the /components option.
/removeall	Removes all installed core components.
/xenapp	Installs XenApp. If this option is omitted, XenDesktop is installed.
/configure_firewall	Opens all ports in the Windows firewall needed by components being installed, if the Windows Firewall Service is running, even if the firewall is not enabled. If you are using a third-party firewall or no firewall, you must manually open the ports.
/components component [component] ...	(Required.) Comma-separated list of components to install or remove. Valid values are: • CONTROLLER - Controller • DESKTOPSTUDIO - Studio • DESKTOPDIRECTOR - Director • LICENSESERVER - Citrix Licensing • STOREFRONT - StoreFront If this option is omitted, all components are installed (or removed, if the /remove option is also specified).
/installdir directory	Existing empty directory where components will be installed. Default = c:\Program Files\Citrix.
/tempdir directory	Directory that holds temporary files during installation. Default = c:\Windows\Temp.
/nosql	Prevents installation of Microsoft SQL Server Express on the server where you are installing the Controller. If this option is omitted, SQL Server Express will be installed.

Option	Description

For example, the following command installs a XenDesktop Controller, Studio, Citrix Licensing, and SQL Server Express on the server. Ports required for component communications will be opened automatically.

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

The following command installs a XenApp Controller, Studio, and SQL Server Express on the server. Ports required for component communication will be opened automatically.

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

When installing a VDA for use with Remote PC Access, specify only options that are valid on physical machines (not VMs or master images) and for VDAs for Windows Desktop OS.

From the XenDesktop Setup directory on the product media, run the XenDesktopVdaSetup.exe command. The following table describes command options. Unless otherwise noted, options apply to physical and virtual machines, and to VDAs for Windows Desktop OS and VDAs for Windows Server OS.

Option	Description
/h or /help	Displays command help.
/quiet or /passive	No user interface appears during the installation. The only evidence of the installation and configuration process is in Windows Task Manager. If this option is omitted, the graphical interface launches.
/logpath path	Log file location. The specified folder must already exist; the installer does not create it. Default = "%TEMP%\CitrixXenDesktop Installer"
/noreboot	Prevents a restart after installation. The VDA will not be fully available for use until after a restart.
/remove	Removes the components specified with the /components option.
/removeall	Removes all installed VDA components.
/reconfig	Customizes previously-configured VDA settings when used with the /portnumber, /controllers, or /enable_hdx_ports options. If you specify this option without also specifying the /quiet option, the graphical interface for customizing the VDA launches.
/portnumber port	(Valid only if the /reconfig option is specified.) Port number to enable for communications between the VDA and the Controller. The previously-configured port is disabled, unless it is port 80.
/components component[component]	Comma-separated list of components to install or remove. Valid values are: • VDA - installs the VDA • PLUGINS - installs the Citrix Receiver for Windows (CitrixReceiver.exe) If this option is omitted, all components are installed.
/installdir directory	Existing empty directory where components will be installed. Default = c:\Program Files\Citrix.
/tempdir directory	Directory to hold temporary files during installation. (This option is not available in the graphical interface.) Default = c:\Windows\Temp.
/site_guid guid	Globally Unique Identifier of the site Active Directory Organizational Unit (OU). This associates a virtual desktop with a Site when you are using Active Directory for discovery (auto-update is the recommended and default discovery method). The site GUID is a site property displayed in Studio. Do not specify both the /site_guid and /controllers options.
/controllers "controller [controller] [...]"	Space-separated Fully Qualified Domain Names (FQDNs) of Controllers with which the VDA can communicate, enclosed in quotation marks. Do not specify both the /site_guid and /controllers options.
/xa_server_location url	URL of the server for Windows server applications.
/enable_remote_assistance	Enables Windows Remote Assistance for use with Director. If you specify this option, Windows opens TCP port 3389 in the firewall, even if you omit the /enable_hdx_ports option.

Option	Description
/enable_hdx_ports	Opens ports in the Windows firewall required by the Controller and features you specified (Windows Remote Assistance, real-time transport, and optimize), if the Windows Firewall Service is detected, even if the firewall is not enabled. If you are using a different firewall or no firewall, you must configure the firewall manually.
/optimize	Enables optimization for VDAs running in a VM on a hypervisor. VM optimization includes disabling offline files, disabling background defragmentation, and reducing event log size. Do not specify this option for Remote PC Access. For more information about the optimization tool, see CTX125874 .
/baseimage	(Valid only when installing a VDA for Windows Desktop OS on a VM.) Enables the use of Personal vDisks with a master image. For more information, see Personal vDisks .
/enable_hdx_3d_pro	Installs the VDA for HDX 3D Pro. For more information, see the HDX 3D Pro documentation.
/enable_real_time_transport	Enables or disables use of UDP for audio packets (Real-Time Audio Transport for audio). Enabling this feature can improve audio performance. Include the /enable_hdx_ports option if you want the UDP ports opened automatically if the Windows Firewall Service is detected.
/masterimage	(Valid only when installing a VDA on a VM.) Sets up the VDA as a master image.
/virtualmachine	(Valid only when installing a VDA on a VM.) Overrides detection by the installer of a physical machine, where BIOS information passed to VMs makes them appear as physical machines.
/nodesktopexperience	(Valid only when installing a VDA for Windows Server OS.) Prevents enabling of the Enhanced Desktop Experience feature. This feature is also controlled with the Enhanced Desktop Experience Citrix policy setting.
/nocitrixwddm	(Valid only on Windows 7 machines that do not include a WDDM driver.) Disables installation of the Citrix WDDM driver.
/servervdi	Installs a VDA for Windows Desktop OS on a supported Windows Server. Omit this option when installing a VDA for Windows Server OS on a Windows Server. Before using this option, see Server VDI . Note: Add the /masterimage option if you are installing the VDA on an image, and will use MCS to create server VMs from that image.
/installwithsecurebootenabled	Allows VDA installation when Secure Boot is enabled. If this option is omitted, a warning displays that Secure Boot must be disabled to successfully install a VDA.
/exclude "Personal vDisk","Machine Identity Service"	(Valid only when upgrading from an earlier 7.x VDA version on a physical machine.) Excludes Personal vDisk and Machine Identity Service from the upgrade. For advanced use of this option, see CTX140972 .

For example, the following command installs a VDA for Windows Desktop OS and Citrix Receiver to the default location on a VM. This VDA will be used as a master image. The VDA will register initially with the Controller on the server named 'Contr-Main' in the domain 'mydomain,' and will use Personal vDisks, the optimization feature, and Windows Remote Assistance.

```
XenDesktop SetupXenDesktopVdaSetup.exe /quiet /components
vda,plugins /controllers "Contr-Main.mydomain.local" /enable_hdx_ports /optimize
/masterimage /baseimage /enable_remote_assistance
```

The following command installs a VDA for Windows Desktop OS and Citrix Receiver to the default location on an office PC that will be used with Remote PC Access. The machine will not be restarted after the VDA is installed; however, a restart is required before the VDA can be used. The VDA will register initially with the Controller on the server named 'Contr-East' in the domain 'mydomain,' and will use UDP for audio packets. HDX ports will be opened if the Windows Firewall service is detected.

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

As an alternative to using the full-product ISO to install VDAs, you can use a standalone VDA installation package. For details, see [Install VDAs using the standalone package](#).

After you install a VDA, you can customize several settings. From the \x64\XenDesktop Setup directory on the product media, run the XenDesktopVdaSetup.exe command, using one or more of the following options, which are described above.

- /reconfigure - this option is required when customizing a VDA
- /h or /help
- /quiet
- /noreboot
- /controllers
- /portnumber port

- /enable_hdx_ports

Notes:

- Deploying the Universal Print Server on 32-bit operating systems is not supported.
- User authentication during printing operations requires the Universal Print Server to be joined to the same domain as the Remote Desktop Services VDA.
- The UPClient component is part of the VDA installation; you do not need to manually install the client component.

To install the Universal Print Server using the command line:

- On a Windows 2008 R2 SP1, Windows Server 2012, or Windows Server 2012 R2 print server, run **UpsServer_x64.msi** from the \x64\Universal Print Server\ directory of the LTSR image.

If you install the Universal Print Server using the command line, we recommend that you add the command option, /ENABLE_CEIP set to 1, to opt in to the [Citrix Customer Experience Improvement Program](#) (CEIP).

For example:



```
msiexec /i UpsServer.msi ENABLE_CEIP=1
```

When you opt in, anonymous statistics and usage information is sent to Citrix to help improve the quality and performance of our products.

创建站点

May 04, 2016

站点是您赋予产品部署的名称。它包含 Delivery Controller、其他核心组件、VDA、虚拟资源连接（如果使用）以及您创建和管理的计算机目录和交付组。站点可以与地理位置对应，但不一定必须对应。可以在安装组件之后创建计算机目录和交付组之前创建站点。

准备

下表介绍了在 Studio 中启动站点创建向导之前需要完成的任务以及需要注意的事项。

说明

确定要创建的站点类型：

应用程序和桌面交付站点 - 如果选择创建应用程序和桌面交付站点，则可以进一步选择创建完整部署站点（推荐）还是空站点。（空站点仅进行部分配置，通常由高级用户创建。）

Remote PC Access 站点 - 允许指定用户通过安全连接远程访问办公室 PC。如果您要使用 Remote PC Access 局域网唤醒功能，请完成 [Configuration Manager and Remote PC Access 局域网唤醒](#)中所述的任务。

如果立即创建应用程序和桌面交付部署，可在以后添加 Remote PC Access 部署。反之，如果此时创建 Remote PC Access 部署，则可以稍后添加完整部署。

站点创建包括创建站点配置数据库。请务必在创建站点之前安装 SQL Server 软件。

您必须是本地管理员和域用户，才能创建数据库。此外，您还必须具有 SQL Server 权限，或者能够生成脚本，供您的数据库管理员运行。

权限 – 设置数据库时，您需要具有以下权限；可以明确配置这些权限，也可以通过 Active Directory 组成员身份获取：

操作	用途	服务器角色	数据库角色
创建数据库	创建合适的空数据库	dbcreator	
创建架构	创建所有服务特定的架构，并将第一个 Controller 添加到站点	securityadmin *	db_owner
添加 Controller	将 Controller（除第一个外）添加到站点	securityadmin *	db_owner
添加 Controller（镜像服务器）	将 Controller 登录信息添加到当前位于镜像 XenDesktop 数据库的镜像角色中的数据库服务器	securityadmin *	
更新架构	应用架构更新或修补程序		db_owner

* 虽然在技术层面的限制更加严格，但实际上应将 securityadmin 服务器角色视为等同于 sysadmin 服务器角色。

✔	说明	用途	服务器角色	数据库角色
	使用 Studio 执行这些操作时，用户帐户必须属于 sysadmin 服务器角色的成员。 如果您的 Studio 用户凭据不包括这些权限，系统会提示您使用 SQL Server 用户凭据。 脚本 - 如果您的数据库服务器已被锁定并且您没有所需的 SQL Server 权限，则可以使用站点创建向导生成两个数据库脚本：一个用于设置数据库，另一个用于镜像环境。请求生成脚本之后，将生成的脚本提供给数据库管理员（或者具有所需 SQL Server 权限的人员）以在数据库服务器上运行，如果需要，还可以在镜像数据库上运行。执行脚本并成功创建数据库后，可以完成创建站点。			
	假设您要使用免费试用 30 天并允许您以后添加许可证文件的许可证，或者要使用现有许可证。可以从站点创建向导内部添加或下载许可证文件。			
	配置虚拟化资源（主机）环境。 如果使用 XenServer： - 请参阅 XenServer 文档。 - 必须提供 VM 超级管理员或更高级别用户的凭据。 - Citrix 建议使用 HTTPS 确保与 XenServer 的通信安全。要使用 HTTPS，必须将 XenServer 上已安装的默认 SSL 证书替换为可信证书机构颁发的证书；请参阅 CTX128656。 - 如果 XenServer 上已启用高可用性，则可以配置高可用性。Citrix 建议选择池中的所有服务器，以便在池主服务器出现故障时允许与 XenServer 实现通信。可以从已添加主机的“编辑高可用性”中选择。 - 如果 XenServer 支持 vGPU，您还可以选择 GPU 类型和组。显示内容将指示所选项是否具有专用 GPU 资源。 如果使用 VMware，请参阅此产品的文档和准备虚拟化环境：VMware。 如果使用 Hyper-V，请参阅此产品的文档和准备虚拟化环境：Microsoft System Center Virtual Machine Manager。 确定使用 Machine Creation Services (MCS) 还是其他工具来在虚拟化资源上创建 VM。 确定要使用共享存储还是本地存储。共享存储可以通过网络获取。如果使用共享存储，则可以使用 IntelliCache 降低存储设备的负载。有关详细信息，请参阅将 IntelliCache 用于 XenServer 连接。 确定是否要使用个人虚拟磁盘以及要使用共享存储还是本地存储。个人虚拟磁盘可以与 VM 使用相同或不同的存储。 如果在云环境中安装了产品组件，则在配置首个连接时，需要使用 API 密钥和密钥值。可以先从 AWS 或 CloudPlatform 导出包含这些值的密钥文件，再将其导入站点创建向导。 为云部署创建站点时，也需要使用在 AWS 中配置的地理区域、可用区域、VPC 名称、子网地址、域名、安全组名称及凭据。			
	确定是否使用 App-V 发布，并根据需要配置这些资源。			
	须知事项： - 创建 Remote PC Access 站点时： - 创建一个名为 Remote PC Access Machines 的计算机目录和一个名为 Remote PC Access Desktops 的交付			

✓	说明 组。 ● 必须指定用户或用户组；不存在自动添加所有用户的默认操作。
	● 可以启用局域网唤醒功能（电源管理），并指定 Microsoft System Center Configuration Manager (ConfigMgr) 地址和凭据以及连接名称。 ● 创建站点的用户将成为完全权限管理员；有关详细信息，请参阅委派管理。 ● 创建空数据库时，该数据库具有除以下属性外的默认属性： ● 排序规则设置为 Latin1_General_100_CI_AS_KS（其中，Latin1_General 因国家/地区而异，例如 Japanese_100_CI_AS_KS）。如果数据库创建期间未指定此排序规则设置，后续在数据库中创建服务架构将失败，并显示一条类似以下内容的错误消息：: schema requires a case-insensitive database (: 架构要求数据库不区分大小写)。（手动创建数据库时，可以使用任意排序规则，前提是该数据库区分大小写、区分重音、区分假名类型；排序规则名称通常以 _CI_AS_KS 结尾。） ● 恢复模式设置为“简单”。要用作镜像数据库，请将恢复模式更改为“完整”。 ● 创建站点配置数据库时，此数据库还存储由 Configuration Logging Service 记录的配置更改，以及由 Monitoring Service 使用并通过 Citrix Director 显示的趋势和性能数据。如果您使用这些功能并存储超过七天的数据，Citrix 建议您在创建站点后，为配置日志记录数据库和监视数据库（称为辅助数据库）指定不同的位置。 ● 命名监视数据库或包含监视数据库的站点配置数据库时，使用包含空格的名称会导致访问数据库时出错。有关详细信息，请参阅 CTX200325。 ● 在站点创建向导的结尾，系统会询问您是否参与 Citrix 客户体验改善计划。加入此计划时，匿名统计信息和使用情况信息将发送到 Citrix；有关详细信息，请参阅关于 Citrix 客户体验改善计划。

创建

如果 Studio 尚未打开，请启动 Studio。选择从中间窗格创建站点后，请指定以下内容：

- 站点类型和站点名称。
- 数据库信息。如果您在 Controller 安装期间选择了安装默认 SQL Server Express 数据库，则某些信息已经提供。如果您使用的数据库服务器安装在其他服务器上，请输入该数据库服务器和名称：

数据库类型	输入内容	使用此数据库配置
独立或镜像	服务器名称	使用默认实例，SQL Server 使用默认端口。
	服务器名称\实例名称	使用指定的实例，SQL Server 使用默认端口。
	服务器名称,端口号	使用默认实例，SQL Server 使用自定义端口。（需要逗号。）
其他	群集名称	群集数据库。
	可用性组侦听器	始终开启的数据库。

单击下一步后，系统将发出警报，说明服务无法连接到数据库，此时请指示需要 Studio 进行创建。如果您没有编辑数据库的权限，请使用生成数据库脚本。必须先运行这些脚本，才能完成站点创建。

- 许可证服务器地址，格式为 name:[port]，其中 name 为完全限定的域名 (FQDN)、NetBIOS 或 IP 地址；建议的格式为 FQDN。如果忽略端口号，则默认为 27000。必须成功连接到许可证服务器才能继续操作。
- （仅限 Remote PC Access 站点。）电源管理信息（包括 ConfigMgr 连接信息）。

- 虚拟化资源的连接信息及存储信息。如果使用的不是资源，或者将使用 Studio 管理专用刀片式服务器 PC 上所托管的用户桌面，请将连接类型选择为无。
- App-V 管理和 App-V 发布服务器信息。
- （仅限 Remote PC Access 站点。）用户和计算机帐户信息。
 - 用户信息。单击添加用户。选择用户和用户组，然后单击添加用户。
 - 计算机帐户信息。单击添加计算机帐户。选择计算机帐户，然后单击添加计算机帐户。单击添加 OU。选择域和组织单位，然后指出是否应包含子文件夹中的项目。单击添加 OU。

测试站点配置

可以查看站点测试结果的 HTML 报告。要运行测试，请执行以下操作：

1. 从 Studio 中，单击左侧窗格顶部的 Studio (<站点名称>) 条目。
2. 在中间窗格中，单击测试站点。

使用脚本安装或删除 Virtual Delivery Agent

May 04, 2016

安装介质包含用于在 Active Directory 中安装、升级或删除计算机组的 Virtual Delivery Agent (VDA) 的示例脚本。您也可以将脚本应用于单个计算机，并用来维护 Machine Creation Services 和 Provisioning Services 所使用的主映像。

所需访问权限：

- 脚本需要对 VDA 安装命令所在的网络共享拥有“所有人可读”访问权限。安装命令是完整产品 ISO 中的 XenDesktopVdaSetup.exe，或独立安装程序中的 VDAWorkstationSetup.exe 或 VDAServerSetup.exe。
- 日志记录详细信息存储在本地计算机上。如果您还要集中记录结果以供查看和分析，则脚本需要对相应网络共享拥有“所有人读/写”访问权限。

要检查运行脚本的结果，请查看中央日志共享。捕获的日志包括脚本日志、安装程序日志及 MSI 安装日志。每次的安装或删除尝试都记录在带时间戳的文件夹中。文件夹标题通过前缀 PASS 或 FAIL 来指示操作是否成功。您可以使用标准目录搜索工具在中央日志共享中快速查找失败的安装或删除，而不是在目标计算机上进行本地搜索。

重要：开始执行任何安装之前，请阅读并完成[准备安装](#)中的任务。

使用脚本安装或升级 VDA

1. 从安装介质上的 \Support\AdDeploy\ 获取示例脚本 InstallVDA.bat。Citrix 建议您先备份原始脚本，再对其进行自定义。
2. 编辑脚本：
 - 指定要安装的 VDA 的版本：SET DESIREDVERSION。例如，在版本 7 中，可以将版本指定为 7.0；可以在 ProductVersion.txt file 中的安装介质上找到其完整值（如 7.0.0.3018）；但是，无需完全匹配。
 - 指定要从中调用安装程序的网络共享位置。指向布局的根目录（树结构的最高点）：将会在脚本运行时自动调用相应的安装程序版本（32 位或 64 位）。例如：SET DEPLOYSHARE=\\fileserver1\share1。
 - 也可以指定用于存储集中式日志的网络共享位置。例如：SET LOGSHARE=\\fileserver1\log1）。
 - 按照[使用命令行安装](#)中的说明指定 VDA 配置选项。默认情况下，脚本中包含 /quiet 和 /noreboot 选项，并且需要这些选项：SET COMMANDLINEOPTIONS=/QUIET /NOREBOOT。
3. 通过组策略启动脚本，将脚本分配给 Active Directory 中计算机所在的 OU。此 OU 应仅包含要安装 VDA 的计算机。重新启动 OU 中的计算机后，脚本在所有这些计算机上运行，在具有受支持操作系统的各计算机上安装 VDA。

使用脚本删除 VDA

1. 从安装介质上的 \Support\AdDeploy\ 获取示例脚本 UninstallVDA.bat。Citrix 建议您先备份原始脚本，再对其进行自定义。
2. 编辑脚本。
 - 指定要删除的 VDA 的版本：SET CHECK_VDA_VERSION。例如，在版本 7 中，可以将版本指定为 7.0；可以在 ProductVersion.txt file 中的安装介质上找到其完整值（如 7.0.0.3018）；但是，无需完全匹配。
 - 也可以指定用于存储集中式日志的网络共享位置。
3. 通过组策略启动脚本，将脚本分配给 Active Directory 中计算机所在的 OU。此 OU 应仅包含要删除 VDA 的计算机。重新启动 OU 中的计算机后，脚本在所有这些计算机上运行，从各计算机中删除 VDA。

故障排除

脚本将生成说明脚本执行进度的内部日志文件。在开始对计算机进行部署的几秒内，脚本会将 Kickoff_VDA_Startup_Script 日志复制到中央日志共享，以便您可以确认整个过程正在运行。如果此日志未按预期复制到中央日志共享，您可以通过检查本地计算机进一步执行故障诊断：脚本将两个调试日志文件放在各计算机的 %temp% 文件夹中，以便尽早进行故障排除：

- Kickoff_VDA_Startup_Script_.log
- VDA_Install_ProcessLog_.log

查看这些日志的内容，以确保该脚本：

- 按预期运行。
- 正确检测目标操作系统。
- 已正确配置为指向 DEPLOYSHARE 共享的 ROOT（包含名为 AutoSelect.exe 的文件）。
- 能够对 DEPLOYSHARE 和 LOG 共享进行身份验证。

使用独立的软件包安装 VDA

Nov 02, 2016

As an alternative to using the full-product XenApp or XenDesktop ISO to install Virtual Delivery Agents (VDAs), you can use a standalone VDA installation package. The smaller package more easily accommodates deployments using Electronic Software Delivery (ESD) packages that are staged or copied locally, have physical machines, or have remote offices.

The standalone VDA package is intended primarily for deployments that use command-line (silent) installation - it supports the same command line parameters as the XenDesktopVdaSetup.exe command, which is used by the full-product installer. The package also offers a graphical interface that is very similar to the VDA installer on the full-product ISO.

There are two self-extracting standalone VDA packages: one for installation on supported server OS machines, and another for supported workstation (desktop) OS machines.

Prerequisites and considerations

The supported operating systems for VDAs, plus other requirements before installation, are listed in [System requirements for XenApp and XenDesktop 7.6](#). See [Prepare to install](#) for details about the information you provide and choices you make during VDA installation.

XenApp and XenDesktop 7.6 FP3. The VDA package automatically deploys prerequisites, if the machine does not already have them; this includes Visual C++ 2008, 2010 and 2013 Runtimes (32-bit and 64-bit) and .NET Framework 4.5.1.

This .NET requirement is skipped by the installer if Windows 10, which includes .NET 4.6, is installed on the desktop where you want to install the VDA.

XenApp and XenDesktop 7.6, 7.6 FP1 and 7.6 FP2. The VDA package automatically deploys prerequisites, if the machine does not already have them; this includes Microsoft Visual C++ 2005, 2008, and 2010 Runtimes (32-bit and 64-bit) and .NET Framework 4.5.1.

When installing on a supported server OS machine, the Remote Desktop Services (RDS) role services are installed and enabled before installing the VDA. Alternatively, you can install the prerequisites yourself before installing the VDA.

Exception: Verify that Windows Server 2008 R2 and Windows 7 machines have at least .NET 3.5.1 installed before you start the VDA installation.

About restarts

- A restart is required at the end of the VDA installation.
- To minimize the number of additional restarts needed during the installation sequence, ensure that .NET Framework 4.5.1 or 4.5.2 is installed before beginning the VDA installation. Also, for Windows Server OS machines, install and enable the RDS role services before installing the VDA. (Other prerequisites do not typically require machine restarts, so you can let the installer take care of those for you.)
- If you do not install prerequisites before beginning the VDA installation, and you specify the /noreboot option for a command line installation, you must manage the restarts. For example, when using automatic prerequisite deployment, the installer will suspend after installing RDS, waiting for a restart; be sure to run the command again after the restart, to continue with the VDA installation.

If you use the graphical interface or the command line interface option that runs the package, the files in the package are extracted to the Temp folder. More disk space is required on the machine when extracting to the Temp folder than when

using the full-product ISO. Files extracted to the Temp folder are not automatically deleted, but you can manually delete them (from C:\Windows\Temp\Ctx-*, where * is a random Globally Unique Identifier) after the installation completes.

Alternatively, use a third party utility that can extract cabinet archives from EXE files (such as 7-Zip) to extract the files to a directory of your choice, and then run the XenDesktopVdaSetup.exe command. In 7.6 FP3, you can use the /extract command with an absolute path. For more information, see How to use in the section below.

If your deployment uses Microsoft System Center Configuration Manager, a VDA installation might appear to fail with exit code 3, even though the VDA installed successfully. To avoid the misleading message, you can wrap your installation in a CMD script or change the success codes in your Configuration Manager package. For more information, see the forum discussion [here](#).

Citrix Display Only Driver in XenApp and XenDesktop 7.6 FP3

The Citrix Display Only Driver (DOD) is the only installed and supported display driver on the XenDesktop Standard VDA on Windows 10.

The Citrix DOD has no GPU assist, even if a GPU or vGPU is present. All rendering is performed by the MS Basic Renderer in the software using the CPU. The Citrix DOD does not support Desktop Composition Redirection (DCR). The Citrix DOD is not installed or supported on XenApp.

How to use

Important: You must either have elevated administrative privileges before starting the installation, or use "Run as administrator."

- 1. Download the appropriate package to the machine where you will be installing the VDA. Citrix account credentials are required to access the download site.

Where are you installing the VDA?	Download this package
On a supported server OS machine	VDAServerSetup.exe
On a supported workstation (desktop) OS machine	VDAWorkstationSetup.exe

For single user, single server OS deployments (for example, delivering Windows Server 2012 to one user for web development), use the VDAWorkstationSetup.exe package. For more information, see [Server VDI](#).

- 2. Install the VDA using the graphical interface or the command line interface.
Remember: You must either have elevated administrative privileges before starting the installation, or use **Run as administrator**.

Using the graphical interface:

Disable User Account Control (UAC), then right-click the downloaded package and choose **Run as administrator**. The installer launches and proceeds through the installation wizard. The restart at the end of the wizard is required before the VDA can be used in a site. (The wizard is the same as the one used in the full-product ISO to install a VDA; you will not encounter anything different.)

Using the command line interface:

You have two options:

- **Extract the files from the package and then run XenDesktopVdaSetup.exe.**

XenApp and XenDesktop 7.6 FP3. To extract the files before installing, use /extract with the absolute path, for example:

```
.\VDAWorkstationSetup.exe /extract %temp%\CitrixVDAInstallMedia
```

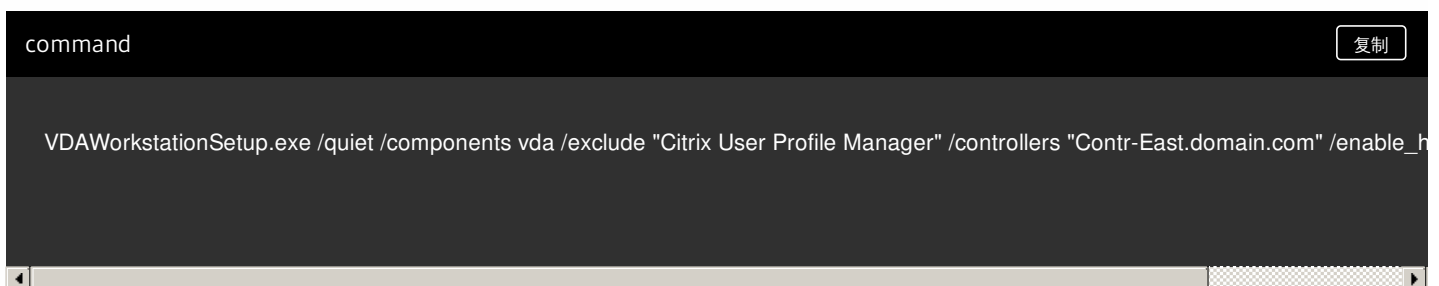
Then, in a separate command, run XenDesktopVdaSetup.exe from the directory containing the extracted content. See [Install using the command line](#) and [CTX140972](#) for parameter information.

XenApp and XenDesktop 7.6, 7.6 FP1 and 7.6 FP2. Extract the files before installing, using a third party utility that can extract cabinet archives from EXE files (such as 7-Zip). Then, in a separate command, run XenDesktopVdaSetup.exe from the directory containing the extracted content. See [Install using the command line](#) and [CTX140972](#) for parameter information.

- **Run the package.**

Run the downloaded package as if it was the XenDesktopVdaSetup.exe command in everything except its name. See [Install using the command line](#) and [CTX140972](#) for parameter information.

For example, the most common installation command used for Remote PC Access installs a VDA on a physical office PC, without installing Citrix Receiver or Citrix Profile Manager. The machine will not automatically be restarted after the VDA is installed; however, a restart is required before the VDA can be used. The VDA will register initially with the Controller on the server named 'Contr-East'. Ports will be opened if the Windows Firewall Service is detected.



The screenshot shows a command prompt window with a dark background. The title bar says 'command' and there is a '复制' (Copy) button in the top right corner. The command entered is: `VDAWorkstationSetup.exe /quiet /components vda /exclude "Citrix User Profile Manager" /controllers "Contr-East.domain.com" /enable_h`

注意

Excluding Citrix Profile management from the installation (Using the /exclude "Citrix User Profile Manager" option) will affect monitoring and troubleshooting of VDAs with Citrix Director. On the User details and EndPoint pages, the Personalization panel and the Logon Duration panel will fail. On the Dashboard and Trends pages, the Average Logon Duration panel will display data only for machines that have Profile management installed.

Even if you are using a third party user profile management solution, it is recommended that you install and run the Citrix Profile management Service to avoid loss of monitoring and troubleshooting in Citrix Director (enabling the Citrix Profile management Service is not required).

计算机目录

May 04, 2016

物理机或虚拟机的集合作为一个称为会话计算机目录的实体进行管理。很多部署都会在其主机上创建一个主映像或模板，然后在计算机目录中以此主映像或模板作为 Citrix 工具（例如 Machine Creation Services 或 Provisioning Services）的指南来从此映像/模板创建 VM。目录还可以包含物理机。

创建计算机目录之后，将会自动运行测试以确保计算机目录配置正确。测试完成后，可以查看测试报告。以后，还可以根据需要从 Studio 导航窗格中的 Citrix Studio 站点名称运行测试。

测试完成后，请创建[交付组](#)。

创建计算机目录

May 04, 2016

如果要使用 Citrix 工具（Machine Creation Services 或 Provisioning Services）为部署创建 VM，请在主机虚拟机管理程序上准备主映像或模板。然后，创建计算机目录。

确保主机的处理器、内存和存储空间足以支持要创建的计算机数。

准备主映像

主映像包含操作系统、非虚拟化应用程序、VDA 和其他软件。VM 是根据之前创建的主映像计算机目录中创建的，它指定何时创建目录。

须知事项：

- 主映像也称为克隆映像、黄金映像或基础映像。
- 云部署使用模板而非主映像。参阅模板指南
 - 在 Amazon Web Services 中，参阅 [Deploy XenApp and XenDesktop 7.5 and 7.6 with Amazon VPC](#)（使用 Amazon VPC 部署 XenApp 和 XenDesktop 7.5 和 7.6）
 - 在 Citrix CloudPlatform 中，请参阅 [XenApp and XenDesktop concepts and deployment on CloudPlatform](#)（XenApp 和 XenDesktop 概念和在 CloudPlatform 上的部署）。

使用 Provisioning Services 时，可以使用主映像或物理计算机作为主目标设备。

- Remote PC Access 计算机目录不使用主映像。
- 使用 Machine Creation Services 时的 Microsoft KMS 激活注意事项：
 - 如果您的部署包括采用 XenServer 6.1 或 6.2、vSphere 或 Microsoft System Center Virtual Machine Manager 主机的 7.x VDA，则无需手动重新部署 Microsoft Windows 或 Microsoft Office。
 - 如果您的部署包括采用 XenServer 6.0.2 主机的 5.x VDA，请参阅 [CTX128580](#)。

重要：如果正在使用 Provisioning Services 或 Machine Creation Services，请勿在主映像上运行 Sysprep。

1. 使用虚拟机管理程序的管理工具创建新的主映像，然后安装操作系统以及所有服务包和更新。
vCPU 数和内存量此时并不重要，因为您在创建计算机目录时可以更改这些值。但是，请确保配置桌面和应用程序所需的硬盘空间量，因为该值不能后续更改或在目录中更改。
2. 确保硬盘连接在设备位置 0 处。大多数标准主映像模板在默认情况下都会配置此位置，但有些自定义模板可能不配置。
3. 在主映像上安装和配置以下软件：
 - 虚拟机管理程序的集成工具（如 XenServer Tools、Hyper-V Integration Services 或 VMware 工具）。如果您忽略此步骤，应用程序和桌面可能无法正常运行。
 - VDA for Windows Server OS 或 VDA for Windows Desktop OS（Citrix 建议安装最新版本以允许访问最新功能）。安装期间，启用通过重新配置某些 Windows 功能来提高性能的优化选项。
 - 所需的第三方工具（例如防病毒软件或电子软件分发代理）。使用适用于用户和计算机类型的设置配置服务，如 Windows Update。
 - 未虚拟化的第三方应用程序。Citrix 建议虚拟化应用程序，因为这样在添加或重新配置应用程序后无需更新主映像，进而显著降低成本。此外，减少安装的应用程序数量还可以减小主映像硬盘的大小，从而节约存储成本。
 - 具有建议设置的 App-V 客户端（如果计划发布 App-V 应用程序）。
 - 使用 Machine Creation Services 且要本地化 Microsoft Windows 时，请安装区域设置和语言包。在置备期间，如果已创建快照，则已置备的 VM 使用已安装的区域设置和语言包。
4. 使用 Provisioning Services 时，为主目标设备中的虚拟磁盘创建一个 VHD 文件，然后再将该主目标设备加入到域中。

5. 将主映像加入到桌面和应用程序所属的域中，并确保主映像将在将要创建计算机的主机上可用。
6. Citrix 建议您创建并命名主映像的快照，以便以后能识别该快照。如果您在创建计算机目录时指定主映像而非快照，Studio 将创建一个快照，但您无法对其进行命名。

为 XenServer 上支持 GPU 功能的计算机准备主映像 - 针对宿主基础结构使用 XenServer 时，支持 GPU 功能的计算机需要专用主映像。这些 VM 要求使用支持 GPU 的视频卡驱动程序，且必须对 VM 进行配置，使其与使用 GPU 进行操作的软件结合使用。

1. 在 XenCenter 中，创建一个具有标准 VGA、网络 and vCPU 的 VM。
2. 更新 VM 配置以启用 GPU 使用（直通或 vGPU）。
3. 安装支持的操作系统并启用 RDP。
4. 安装 XenServer Tools 和 NVIDIA 驱动程序。
5. 关闭虚拟网络计算 (Virtual Network Computing, VNC) 管理控制台以优化性能，然后重新启动 VM。
6. 系统将提示您使用 RDP。使用 RDP 安装 VDA，然后重新启动 VM。
7. 或者，创建 VM 的一个快照作为其他 GPU 主映像的基线模板。
8. 使用 RDP，安装在 XenCenter 中配置并使用 GPU 功能的客户特定应用程序。

创建计算机目录

启动计算机目录创建向导之前，请查看以下过程以了解您需要做出的选择以及需要提供的信息。启动向导时，根据您的环境以及所做选择，某些项可能不显示，或者可能具有不同的标题。

在 Studio 中：

- 如果您已创建站点但尚未创建计算机目录，Studio 将指导您进入正确的起始位置以创建计算机目录。
- 如果您已创建计算机目录且要创建另一目录，请在 Studio 导航窗格中选择计算机目录，然后在“操作”窗格中选择 Create Machine Catalog（创建计算机目录）。

向导将指导您完成下述项。

• 操作系统

每个目录都只包含一种类型的计算机：

- Windows 服务器操作系统 – Windows 服务器操作系统目录提供可由多个用户共享的桌面和应用程序。
- Windows 桌面操作系统 – Windows 桌面操作系统目录提供分配给单独用户的桌面和应用程序。
- Remote PC Access – Remote PC Access 目录为用户提供对其物理办公桌面计算机的远程访问权限。Remote PC Access 不需要 VPN 提供安全性。

Amazon Web Services (AWS) 仅支持服务器操作系统计算机目录（和服务器 VDI，请参阅[服务器 VDI](#)），不支持桌面操作系统或 Remote PC Access 目录。

• 计算机管理

指示目录中的计算机是否通过 Studio 进行电源管理：

- 计算机通过 Studio 进行电源管理或通过云环境进行置备（例如，VM 或刀片式 PC）。仅在已配置虚拟机管理程序或云环境连接时，此选项才可用。您在创建站点时可能已配置连接。如果尚未配置，可在以后创建一个新连接，然后编辑计算机目录。
- 计算机不通过 Studio 进行电源管理（例如物理机）。

指出要使用哪种工具部署计算机：

- Machine Creation Services (MCS) – 使用主映像或模板创建和管理虚拟机。
 - MCS 不可用于物理机。
 - 云环境中的计算机目录使用 MCS。
- Provisioning Services – 将目标设备作为一个设备集合进行管理。从主目标设备进行映像的 Provisioning Services 虚拟磁盘交付桌面和应用程序。
- 其他 – 已在数据管理中心管理计算机的工具。Citrix 建议您使用 Microsoft System Center Configuration Manager 或其他

第三方应用程序，以确保目录中的计算机一致。

- 桌面体验

对于包含将用于交付桌面的桌面操作系统计算机的计算机目录：

- 指定用户是在每次登录时连接到新（随机）桌面，还是每次都连接到同一（静态）桌面。
- 如果用户连接到同一桌面，请指定他们在桌面上进行的任何更改将发生的情况。可以将更改保存到单独的个人虚拟磁盘或用户的本地 VM 磁盘，也可以放弃更改。（如果选择将更改保存到单独的个人虚拟磁盘，请稍后在向导中指定驱动器盘符和大小）。

- 主映像或计算机模板

选择之前创建的主映像（非云）或计算机模板（云）。谨记：如果正在使用 Provisioning Services 或 Machine Creation Services，请勿在主映像上运行 Sysprep。

- 安全性

（云环境）为 VM 选择一个或多个安全组；这些组仅在可用区域支持安全组时显示。选择计算机将要使用共享硬件还是帐户专用硬件。

- 虚拟机或设备集合或 VM 和用户

指定要创建的虚拟机数。可以选择每台计算机将具有的虚拟 CPU 数和内存量（以 MB 为单位）。每个 VM 都将具有 32 GB 硬盘；此值在主映像中设置，无法在目录中更改。

如果您之前指示应将桌面的用户更改保存到单独的个人虚拟磁盘上，请指定其大小（以 GB 为单位）和驱动器盘符。

如果计划使用多个网络接口卡 (NIC)，请将虚拟网络与每个卡相关联。例如，可以分配一个卡用于访问特定的安全网络，另一个卡用于访问更为常用的网络。也可以在此向导中添加或删除 NIC。

- 计算机帐户

（Remote PC Access 目录）指定要添加的对应于用户或用户组的 Active Directory 计算机帐户或组织单位 (OU)。

您可以选择之前配置的电源管理连接，也可以选择不使用电源管理。如果要使用电源管理但尚未配置合适的连接，可以稍后创建该连接，然后编辑计算机目录以更新电源管理设置。

- 计算机帐户

目录中的每台计算机都必须具有一个相应的 Active Directory 计算机帐户。指示是创建新帐户还是使用现有帐户，并指出这些帐户的位置。

如果您使用现有帐户，请确保对要创建的计算机具有足够的未使用计算机帐户。

可以浏览 Active Directory 找到现有帐户，也可以导入列出帐户名的 .csv 文件。导入的文件内容必须使用以下格式：

```
[ADComputerAccount] ADcomputeraccountname.domain ...
```

对于包含物理机或现有计算机的目录，选择或导入现有帐户，并将每台计算机同时分配给 Active Directory 计算机帐户和用户帐户。

对于使用 Provisioning Services 创建的计算机，目标设备的计算机帐户的管理方式不同；请参阅 Provisioning Services 文档。

另外，为创建的计算机指定帐户命名方案 – 方案中的井号标记 (#) 表示序列号或字母，它们将包括在您提供的附加名称文本中。

- 名称和说明

在创建向导的最后一页上，指定计算机目录的名称和说明。此信息显示在 Studio 中。

管理计算机目录

May 04, 2016

对于随机计算机目录，可以通过对主映像应用全局更改（例如 Windows 更新、防病毒软件更新、操作系统升级或配置更改）来维护用户的桌面。然后，可以将计算机目录修改为使用更新后的主映像，这样，用户便可在下次登录时收到更新后的桌面。可以在一次操作中对大量用户进行重要更改。

对于静态和 Remote PC Access 计算机目录，必须使用第三方软件分发工具以单独或集中方式在 Studio 外部管理用户桌面更新。对于通过 Provisioning Services 创建的计算机，用户桌面更新通过虚拟磁盘进行传播。

Citrix 建议您在进行更新前先保存主映像的副本或快照。数据库会保留每个计算机目录中使用主映像的历史记录。不要对主映像进行删除、移动或重命名。如果已部署到用户桌面的更新出现问题，可以将计算机目录还原为使用上一版本的主映像，从而最大程度缩短用户停机时间。

添加计算机

开始之前：

- 确保虚拟化主机具有足够多的处理器、内存和存储空间来容纳更多的计算机。
- 确保有足够多的未使用 Active Directory 计算机帐户。如果使用现有帐户，请注意，可以添加的计算机数受可用帐户数的限制。
- 如果使用 Studio 来为更多计算机创建 Active Directory 计算机帐户，还必须具有相应的域管理员权限。

1. 在 Studio 导航窗格中选择计算机目录。
2. 选择计算机目录，然后在“操作”窗格中选择添加计算机。
3. 选择要添加的虚拟机数。
4. 如果指示应创建新的 Active Directory 帐户（如果现有帐户不足以应对要添加的 VM 数，需要此步骤）：
 - 选择将创建帐户的域和位置。
 - 指定帐户命名方案，使用哈希值标记指示连续数字或字母将要显示的位置（名称不得以数字开头）。例如，命名方案 PC-Sales-##（可以选择 0-9）将生成名为 PC-Sales-01、PC-Sales-02、PC-Sales-03 等的计算机帐户。

如果指出应使用现有 Active Directory 帐户：

- 浏览到相应帐户，或单击“导入”并指定一个包含帐户名称的 .csv 文件。确保所有要添加的计算机都有足够的帐户。
- 由于 Studio 将管理这些帐户，因此应允许 Studio 重置所有帐户的密码或者指定帐户密码（所有帐户的密码必须相同）。

系统会将计算机创建过程作为后台进程来执行，如果要创建大量计算机，则创建过程会非常冗长。即使关闭 Studio，计算机创建过程也会继续执行。

更改计算机目录说明或更改 Remote PC Access 设置

1. 在 Studio 导航窗格中选择计算机目录。
2. 选择目录，然后在“操作”窗格中选择编辑计算机目录。
3. （仅限 Remote PC Access 目录）在电源管理页面上，可以更改 Remote PC Access 目录的电源管理设置并选择电源管理连接。在组织单位页面上，添加或删除 OU。
在说明页面上，更改计算机目录说明。

重命名计算机目录

1. 在 Studio 导航窗格中选择计算机目录。
2. 选择目录，然后在“操作”窗格中选择重命名计算机目录。
3. 输入新名称。

删除计算机目录

删除计算机目录之前，请确保：

- 所有用户均已注销且没有仍在运行的已断开连接的会话。
- 目录中的所有计算机均已打开维护模式，这样所有计算机均已关闭。
- 目录未与交付组相关联。

1. 在 Studio 导航窗格中选择计算机目录。
2. 选择目录，然后在“操作”窗格中选择删除计算机目录。
3. 指明是否应删除目录中的计算机。如果选择删除这些计算机，应指明关联的计算机帐户是应该在 Active Directory 中原样保留、禁用还是删除。

从计算机目录中删除计算机

从目录中删除某一计算机之后，用户将不能再访问该计算机。删除计算机之前，请确保：

- 用户数据已备份或者不再需要。
- 所有用户均已注销。打开维护模式后，用户将无法连接到计算机。
- 桌面未启动或已挂起。

1. 在 Studio 导航窗格中选择计算机目录。
2. 选择计算机目录，然后在“操作”窗格中选择“查看计算机”。
3. 选择一个或多个计算机，然后在“操作”窗格中单击“Turn On Maintenance Mode”（打开维护模式）。
4. 在“操作”窗格中选择“删除”。
5. 选择是否删除要移除的计算机。如果选择删除这些计算机，请选择如何处理关联的 Active Directory 计算机帐户：

在计算机目录中	在 Active Directory 中
保留	不更改
删除	不删除
删除	禁用
删除	删除

管理 Active Directory 计算机帐户

要管理计算机目录中的 Active Directory 帐户，可以：

- 从桌面操作系统计算机目录和服务器操作系统计算机目录中删除 Active Directory 计算机帐户，从而释放未使用的计算机帐户。之后，这些帐户便可用于其他计算机。
- 添加帐户，以便在向此目录添加更多计算机时，有可用的计算机帐户

1. 在 Studio 导航窗格中选择计算机目录。
2. 选择计算机目录，然后在“操作”窗格中选择管理 AD 帐户。
3. 选择是添加还是删除计算机帐户。
 - 如果添加帐户，系统会提示您指定帐户密码的处理方式：重置所有密码还是输入一个适用于所有帐户的密码。如果不知道当前的帐户密码，则可能需要重置密码；必须具有重置密码的权限。如果输入一个密码，帐户在导入时会更改密码。
 - 如果删除帐户，系统会提示您选择是在 Active Directory 中保留、禁用还是删除帐户。

更新主映像

更新主映像，以便将更改应用到计算机目录中使用此主映像创建的所有桌面和应用程序。通过单个主映像对通用环节进行管理，这样即可非常快捷地将系统范围的更改（例如 Windows 更新或配置更改）部署到大量的计算机。

在主机上准备并测试新的/更新后的主映像之后（请参阅[准备主映像](#)），将计算机目录修改为使用此主映像。

请注意以下问题：

- Citrix 建议您在进行更新前先保存主映像的副本或快照。数据库会保留每个计算机目录中使用主映像的历史记录。如果已部署到用户桌面的更新出现问题，可以将计算机目录还原为使用上一版本的主映像，从而最大程度缩短用户停机时间。不要删除、移动或重命名主映像；否则，您无法将计算机目录还原为使用这些主映像。
虽然 Studio 可以创建快照，Citrix 仍建议您使用虚拟机管理程序管理控制台创建快照，然后在 Studio 中选择该快照。使用此方法可以提供有意义的名称及说明，而非自动生成的名称。
 - 对于 GPU 主映像，只能通过 XenServer 的 XenCenter 控制台进行更改。
 - 对于使用 Provisioning Services 的计算机目录，必须发布新的虚拟磁盘才能将更改应用到目录。有关详细信息，请参阅 Provisioning Services 文档。
 - 更新主映像之后，您必须通过 Studio 重新启动计算机，从而使所做的更改生效并可以供用户使用。这可能会自动发生；例如，当用户从桌面注销时，或者，更改也可能随配置的重新启动计划而生效。或者，也可以从 Studio 重新启动计算机。
1. 在 Studio 导航窗格中选择计算机目录。
 2. 选择计算机目录，然后在“操作”窗格中选择更新计算机。
 3. 在主映像页面上，选择主机和新的/更新的主映像。
 4. 在回滚策略页面上，指定将新的/更新的主映像应用于用户计算机的时间：下次关闭时还是立即应用。
 - 如果选择在下次关闭时更新映像，则可以向用户通知更新。
 - 如果选择立即更新映像，则可以指定是同时还是以指定的时间间隔重新启动所有计算机。可以在将用户注销并重新启动计算机之前 1 分钟、5 分钟或 15 分钟向用户发送通知消息。

还原为先前版本的主映像

1. 在 Studio 导航窗格中选择计算机目录。
2. 选择计算机目录，然后在“操作”窗格中选择回滚计算机更新。
3. 指定如何将还原的主映像应用于用户桌面，如上所述。

回滚策略仅适用于需要还原的桌面。如果系统提示将回滚新的/更新的主映像，但有些桌面尚未根据这些主映像进行更新（例如，用户尚未注销的桌面），这些桌面的用户将不会收到任何消息，也不会被强制注销。

升级计算机目录

在将计算机上的 VDA 升级到最新版本之后，可以升级计算机目录。Citrix 建议将所有 VDA 均升级到最新版本，以使其全部能访问最新功能。

注意：如果使用的是安装 Windows XP 或 Windows Vista 的计算机，这些计算机必须使用早期版本的 VDA，并且无法使用最新的产品功能。如果无法将这些计算机升级到当前支持的 Windows 操作系统，Citrix 建议将其保存在一个单独的计算机目录中。有关详细信息，请参阅[运行 Windows XP 或 Windows Vista 的计算机上的 VDA](#)和[混合 VDA 支持](#)。

升级计算机目录之前：

- 如果使用的是 Provisioning Services，请在 Provisioning Services 控制台中升级 VDA 版本。
 - 启动升级后的计算机，使其注册到 Controller 中。这样，Studio 便可以确定计算机目录中的计算机是否需要升级。
1. 在 Studio 导航窗格中选择计算机目录。
 2. 选择相应的计算机目录。下部窗格中的“详细信息”选项卡会显示版本信息。

3. 选择升级目录。

- 如果 Studio 检测到目录需要升级，它会显示一条消息。按照提示进行操作。
- 如果一个或多个计算机无法升级，则消息中会说明原因。Citrix 建议您在升级计算机目录前解决计算机问题，以确保所有计算机均正常运行。

还原计算机目录升级

在还原计算机目录升级之前，如果您使用了 Provisioning Services 来创建计算机目录，请在 Provisioning Services 控制台中更改 VDA 版本。

1. 在 Studio 导航窗格中选择计算机目录。
2. 选择相应的计算机目录。下部窗格中的“详细信息”选项卡会显示版本信息。
3. 选择撤消，然后按照提示进行操作。

交付组

May 04, 2016

交付组是从一个或多个计算机目录中选择的计算机的集合。交付组指定哪些用户可以使用这些计算机，以及可供这些用户使用的应用程序。

首先创建交付组。稍后，您可以更改初始设置并配置其他项。

创建交付组

要创建交付组，请执行以下操作：

1. 在 Studio 导航窗格中选择交付组。
2. 在“操作”窗格中选择创建交付组。向导将指导您完成下述项。

计算机

选择计算机目录并指定要从目录中使用的计算机数。

- 在选定的计算机目录中，必须至少有一个计算机处于未使用状态。
- 一个计算机目录可以在多个交付组中指定；但是，一个计算机只能用于一个交付组。
- 交付组可以使用多个计算机目录；但是，这些目录必须包含相同的计算机类型（服务器操作系统、桌面操作系统或 Remote PC Access）。换言之，不能在交付组或计算机目录中混合使用多个计算机类型。
- 同样，无法创建既包含为静态桌面配置的计算机目录中的桌面操作系统计算机，又包含为随机桌面配置的计算机目录中的计算机的交付组。
- Remote PC Access 计算机目录中的每台计算机都自动与交付组关联。

交付类型

类型指示交付组所提供的内容：仅桌面、仅应用程序还是同时包括桌面和应用程序。具有静态桌面操作系统计算机的交付组无法同时提供桌面和应用程序。

用户

指定能够使用交付组中的应用程序和/或桌面的用户和用户组。

用户类型有两种：已通过身份验证和未经身份验证（后者也称为匿名）。您可以配置两种类型之一或全部。

- **已通过身份验证** - 您按名称指定的用户和组成员必须向 StoreFront 或 Citrix Receiver 提供凭据（如智能卡或用户名和密码），才能访问应用程序和桌面。
- **未经身份验证（匿名）** - 对于包含服务器操作系统计算机的交付组，您可以选中特定的复选框，从而允许用户访问应用程序和桌面而不向 StoreFront 或 Citrix Receiver 提供凭据。例如，当用户通过 kiosk 访问应用程序时，应用程序可能需要凭据，但 Citrix 访问门户和工具则不需要。安装 VDA 时，会创建匿名用户组。
 - 要向未经身份验证的用户授予访问权限，交付组中的每台计算机必须已安装 VDA for Windows Server OS（最低版本为 7.6）。启用未经身份验证的用户时，您必须具有未经身份验证的 StoreFront 存储。
 - 启动会话时，将按需创建未经身份验证的用户帐户，其名称为 AnonXYZ，其中，XYZ 是一个唯一的三位数值。
 - 未经身份验证的用户会话的默认空闲超时为 10 分钟，当客户端断开连接时，这些会话将自动注销。不支持重新连接、客户端之间漫游以及 Workspace Control。

下表介绍了您的选项。

启用访问权限的用户	是否添加/分配用户和用户组？	是否启用“Give access to unauthenticated users”（向未经身份验证的用户授予访问权限）复选框？

启用访问权限的用户 仅限已通过身份验证的用户	是否添加/分配用户 和用户组？	是否启用“Give access to unauthenticated users”（向未经身份验证的用户授予访问权限）复选框？
仅未经身份验证的用户	否	是
已通过身份验证的用户和未经身份验证的用户	是	是

对于包含桌面操作系统计算机的桌面组，您可以在创建交付组之后导入用户数据（用户列表）。请参阅[导入或导出用户列表](#)。

应用程序

一个列表显示在基于主映像或计算机目录中的模板所创建的计算机上或者 App-V 管理服务器上发现的应用程序。选择要添加到交付组的一个或多个应用程序。

也可以手动添加（创建）应用程序。您需要提供可执行文件路径、工作目录、可选命令行参数以及管理员和用户的显示名称。

您可以更改应用程序的属性；有关详细信息，请参阅[更改应用程序属性](#)。

您无法为 Remote PC Access 交付组创建应用程序。

默认情况下，您添加的应用程序位于 Applications 文件夹中。利用文件夹可以更轻松地管理大量应用程序。添加应用程序时，您可以指定不同的文件夹；但是，以后管理文件夹会更加轻松。有关详细信息，请参阅[管理应用程序文件夹](#)。

如果使用相同名称向同一用户发布两个应用程序，请在 Studio 中更改应用程序名称(面向用户) 属性，否则，用户将在 Receiver 中看到重复的名称。

StoreFront

选择或添加交付组中每台计算机上所安装的 Citrix Receiver 将要使用的 StoreFront URL。也可以通过在导航窗格中依次选择配置 > StoreFront 来指定 StoreFront 服务器地址。添加 StoreFront 服务器时会向 URL 结尾添加“/Discovery”。

设置

May 04, 2016

以下文档介绍了如何配置和管理可为交付组指定和更新的大多数设置：

- [应用程序](#)
- [计算机](#)
- [Remote PC Access](#)
- [会话](#)
- [用户](#)

以下信息介绍了这些文档中未涵盖的设置。

更改交付组

将仅应用程序或桌面和应用程序交付组更改为仅桌面交付组之前，请从交付组中删除所有应用程序。

1. 在 Studio 导航窗格中选择交付组。
2. 选择交付组，然后在“操作”窗格中选择编辑交付组。
3. 在交付类型页面上，选择要将交付组更改到的交付类型。

更改基本设置

1. 在 Studio 导航窗格中选择交付组。
2. 选择交付组，然后在“操作”窗格中选择编辑交付组。
3. 在基本设置页面上，您可以更改以下项：

设置	说明
说明	StoreFront 使用且用户能够看到的文本。
“已启用”复选框	是否启用交付组。
每个用户的桌面数	(仅限桌面操作系统计算机) 用户可以同时激活的最大共享桌面数。在第一次使用时指定的部署中，此值指定了用户可为其自身分配的桌面数。
时区	
启用 Secure ICA	使用 SecureICA 确保与交付组中的计算机之间的通信安全，这会加密 ICA 协议（默认级别为 128 位；可使用 SDK 更改该级别）。Citrix 建议在遍历公共网络时使用其他加密方法（如 SSL/TLS 加密）。SecureICA 也不检查数据完整性。

升级交付组

在计算机上升级 VDA 之后，升级交付组。

注意：如果您必须使用早期 VDA 版本，更新的产品功能可能不可用。有关详细信息，请参阅[升级部署](#)。

启动交付组升级之前：

- 如果您使用 Provisioning Services，则在 Provisioning Services 控制台中升级 VDA 版本。
- 启动包含新 VDA 的计算机，以便向 Controller 注册它们。此过程将指示 Studio 交付组中需要升级的内容。

1. 在 Studio 导航窗格中选择交付组。
2. 选择交付组，然后在“操作”窗格中选择升级交付组。

启动升级过程之前，Studio 将指示您无法升级的计算机（如果有）及原因。您随后可以取消升级、解决计算机问题，然后再次启动交付组升级。

交付组升级完成后，您可以通过选择交付组，然后在“操作”窗格中选择撤消，将计算机还原为其先前状态。

计算机

Jan 04, 2017

除非另有说明，否则所有交付组类型（服务器操作系统、桌面操作系统和 Remote PC Access）均支持以下过程。

关闭和重新启动计算机

注意：Remote PC Access 计算机不支持此过程。

1. 在 Studio 导航窗格中选择交付组。
2. 选择交付组，然后在“操作”窗格中选择查看计算机。
3. 选择计算机，然后在“操作”窗格中选择下列项之一（某些选项可能不可用，具体视计算机状态而定）：
 - 强制关闭 — 强行关闭计算机并刷新计算机列表。
 - 重新启动 — 请求关闭操作系统，然后再次启动计算机。如果操作系统无法关闭，计算机将保持其当前状态。
 - 挂起 — 不关闭但暂停计算机并刷新计算机列表。
 - 关闭 — 请求关闭操作系统。

如果计算机在 10 分钟内没有关闭，则会关机。如果 Windows 尝试在关闭期间安装更新，可能面临在更新完成前计算机关闭的风险。

注意：Citrix 建议阻止桌面操作系统计算机用户在会话中选择“关闭”。有关详细信息，请参阅 Microsoft 策略文档。

计算机电源管理

注意：您只能对虚拟桌面操作系统计算机进行电源管理，而不能对物理机（包括 Remote PC Access 计算机）进行电源管理。具有 GPU 功能的桌面操作系统计算机无法挂起，因此关机操作失败。对于服务器操作系统计算机，请参阅[创建重新启动计划](#)。计算机可以处于下列状态之一：

交付组	状态
随机	随机分配并且正在使用
	未分配并且未连接
静态（已分配）	永久分配并且正在使用
	永久分配并且未连接（但已就绪）
	未分配并且未连接

在正常使用期间，静态交付组通常既包括永久分配的计算机，也包括未分配的计算机。最初，所有计算机均未分配（创建交付组时手动分配的计算机除外）。当用户连接时，计算机变为永久分配状态。您可以对这些交付组中的未分配计算机进行全面的电源管理，但对永久分配的计算机却只能进行部分管理。

- **池和缓冲区** - 对于随机交付组和静态交付组中的未分配计算机，池是一组保持为启动状态以供用户连接的未分配（或临时分配）计算机；用户在登录后将立刻获得计算机。池大小（保持为启动状态的计算机数）可按一天中的具体时刻进行配置。（对于静态交付组，请使用 SDK 配置池。）

缓冲区是另外一组未分配的备用计算机，它们在池中的计算机数低于阈值（交付组大小的百分比）时打开。对于大型交付组，超过阈值时可能会打开大量计算机，因此请谨慎规划交付组大小或使用 SDK 调整默认缓冲区大小。

- **电源状态计时器** - 您可以使用电源状态计时器在用户断开连接指定时间后挂起计算机。例如，在非工作时间，计算机将在用户断开连接至少十分钟后自动挂起。除非您配置 SDK 中的 ShutdownDesktopsAfterUse 交付组属性，否则随机计算机或具有个人虚拟磁盘的计算机将在用户注销时自动关闭。
您可以针对工作日和周末以及峰值和非峰值间隔配置计时器。

- **永久分配计算机的部分电源管理** - 对于永久分配的计算机，您可以设置电源状态计时器，但对池或缓冲区则不可。这些计算机在每个高峰期到来时打开，在每个非高峰期到来时关闭，因此您无法像处理未分配计算机那样精细控制用来补偿被占用计算机的可用计算机数。

要对虚拟桌面操作系统计算机进行电源管理，请执行以下操作：

1. 在 Studio 导航窗格中选择交付组。
2. 选择交付组，然后在“操作”窗格中选择编辑交付组。
3. 在电源管理页面上，选择对计算机进行电源管理下拉菜单中的工作日。（默认情况下，工作日是指周一到周五。）
4. 对于随机交付组，在 Machines to be powered on（要启动的计算机）中，选择编辑并指定工作日期间的池大小。然后，选择要启动的计算机数。
5. 在高峰小时数中，设置每天的高峰时段和非高峰期间。
6. 设置工作日高峰时段和非高峰时段的电源状态计时器：
 - 在高峰期间 > 断开连接时中，指定挂起交付组中任何已断开计算机前的延迟时间（分钟），然后选择挂起。
 - 在非高峰期间 > 断开连接时中，指定关闭交付组中任何已注销计算机前的延迟时间，然后选择关闭。此计时器不可用于具有随机计算机的交付组。
8. 在对计算机进行电源管理下拉菜单中选择周末，然后配置周末的高峰时段和电源状态计时器。

使用 SDK 可以执行以下操作：

- 关闭而非挂起计算机以响应电源状态计时器，或者在希望计时器基于注销数而非断开连接数时使用。
- 更改默认的工作日和周末定义。

创建重新启动计划

注意：只能针对服务器操作系统计算机使用重新启动计划。对于桌面操作系统计算机，请参阅[计算机电源管理](#)。

要配置重新启动计划，请执行以下操作：

1. 在 Studio 导航窗格中选择交付组。
2. 选择交付组，然后在“操作”窗格中选择编辑交付组。
3. 在重启计划页面上：
 - 在重新启动计算机下拉列表中，选择重新启动计算机的频率。
 - 在在以下时间重新启动第一个组字段中，指定第一个服务器开始重启过程的小时和分钟（采用 24 小时制）。
 - 在每隔以下时间重新启动其他组下拉列表中，指定是否同时重新启动所有服务器，或重新启动交付组中的每个服务器所允许的时间长度。

例如，假定一个交付组包含五个服务器，为在以下时间重新启动第一个组选择 13:00（下午 1:00），为每隔以下时间重新启动其他组选择 1 小时。持续时间（60 分钟）除以计算机数（5），将得到 12 分钟的重新启动间隔。因此，重新启动时间为下午 1:00、下午 1:12、下午 1:24、下午 1:36 和下午 1:48。这样可允许所有五个计算机在指定时间间隔（1 小时）结束时均完成重新启动。

- 指出在用户注销之前是否以指定间隔向其发送消息。将相对于每个服务器的预计重启时间发送通知，如示例中所述。

不能在 Studio 中执行自动打开或关闭操作。

阻止用户连接到计算机（维护模式）

当您需临时停止计算机的新连接时，可以针对交付组中的一个或所有计算机打开维护模式。您可能会在应用修补程序或使用

管理工具之前执行此操作。

- 当服务器操作系统计算机处于维护模式时，用户可以连接到现有会话，但无法启动新会话。
- 当桌面操作系统计算机（或使用 Remote PC Access 的 PC）处于维护模式时，用户无法连接或重新连接。当前连接仍保持连接状态，直到其断开连接或注销。

1. 在 Studio 导航窗格中选择交付组。
2. 选择交付组。
3. 要针对交付组中的所有计算机打开维护模式，请在“操作”窗格中选择打开维护模式。
要针对一台计算机打开维护模式，请执行以下操作：
 1. 在“操作”窗格中选择查看计算机。
 2. 选择计算机，然后在“操作”窗格中选择打开维护模式。
4. 要针对交付组中的一台或所有计算机关闭维护模式，请按照之前的说明操作，不过，要在“操作”窗格中选择关闭维护模式。

Windows 远程桌面连接 (RDC) 设置还影响服务器操作系统计算机是否处于维护模式。以下任一情况下，维护模式将打开：

- 服务器维护模式设置为“打开”，如上所述。
- RDC 设置为“Don't allow connections to this computer”（不允许连接到这台计算机）。
- RDC 设置为不允许连接到这台计算机，并且“远程主机配置用户登录模式”设置为以下项之一：
 - 允许重新连接，但拒绝新用户登录
 - 允许重新连接，但服务器重新启动后才允许新用户登录。

重新分配计算机（更改用户）

注意：您只能重新分配桌面操作系统计算机，不能重新分配服务器操作系统计算机或通过 Provisioning Services 创建的计算机。

1. 在 Studio 导航窗格中选择交付组。
2. 选择交付组。
3. 要重新分配多台计算机，请执行以下操作：
 1. 在“操作”窗格中选择编辑交付组。
 2. 在计算机分配（用户分配）页面上，选择计算机并指定新用户。
4. 要重新分配一台计算机，请执行以下操作：
 1. 在“操作”窗格中选择查看计算机。
 2. 选择计算机，然后在“操作”窗格中选择更改用户。
 3. 添加或删除用户。

更改每个用户的最大计算机数

1. 在 Studio 导航窗格中选择交付组。
2. 选择交付组，然后在“操作”窗格中选择编辑交付组。
3. 在用户设置页面上，设置每个用户的桌面数值。

使用标记标识计算机

注意：只能对桌面操作系统计算机使用标记。

可以使用标记缩小计算机搜索范围或限制计算机访问。可以添加任意数量、任何长度的标记。

1. 在 Studio 导航窗格中选择交付组。
2. 选择交付组，然后在“操作”窗格中选择查看计算机。
3. 选择计算机。
4. 要添加标记，请在“操作”菜单中选择添加标记，然后输入一个或多个标记（以分号 (;) 分隔）。
要更改或删除标记，请在“操作”菜单中选择编辑标记，然后进行必要的更改。

负载管理

注意：只能对服务器操作系统计算机进行负载管理。

负载管理可测量服务器负载并决定在当前环境条件下选择哪个服务器。其选择的依据包括：

- **服务器维护模式状态** – 仅在维护模式关闭的情况下，才考虑将服务器操作系统计算机用于负载平衡。（有关详细信息，请参阅[阻止用户连接到计算机（维护模式）](#)。）
- **服务器负载指数** – 决定交付服务器操作系统计算机的服务器接收连接的可能性。该指数是负载评估程序的组合：会话数和性能指标（如 CPU、磁盘和内存使用情况）的设置。负载评估程序在负载管理策略设置中指定。
 - 您可以在 Director、Studio 搜索和 SDK 中监视负载指数。
 - 在 Studio 中，“Server Load Index”（服务器负载指数）列默认处于隐藏状态。要显示该列，请选择计算机，通过右键选择列标题，然后选择选择列。在“计算机”目录中，选择负载指数。
 - 在 SDK 中，使用 Get-BrokerMachine cmdlet。

服务器负载指数为 10000 表示服务器处于全负载状态。如果没有其他服务器可用，则用户启动会话时可能会收到一条消息，说明桌面或应用程序当前不可用。

- **并发登录容差策略设置** - 登录到服务器的最大并发请求数。（在 7.5 之前的 XenApp 版本中，此设置等效于负载限制。）

如果所有服务器都等于或高于并发登录容错设置，则会将下一个登录请求分配给挂起登录最少的服务器。如果有多个服务器符合此条件，则会选择负载指数最低的服务器。

有关详细信息，请参阅

— [策略设置参考](#)

。

删除计算机

删除某台计算机会将其从交付组中删除，但不会从交付组所使用的计算机目录中删除。因此，可将计算机分配给其他交付组。

必须先关闭计算机，之后才能将其删除。要在删除计算机时暂时阻止用户连接到该计算机，请先将其置于维护模式，然后再关闭计算机。

请记住，计算机可能包含个人数据，因此将其分配给其他用户之前应小心谨慎。您可能需要重新建立计算机映像。

1. 在 Studio 导航窗格中选择交付组。
2. 选择交付组，然后在“操作”窗格中选择查看计算机。
3. 确保所有计算机已关闭。
4. 在“操作”窗格中选择从交付组中删除。

限制访问计算机

无论使用何种方法，为限制访问交付组中的计算机所做的任何更改都将取代以前的设置。可以执行以下操作：

- 使用委派管理作用域限制管理员的访问权限。可以创建并分配两个作用域，一个允许管理员访问所有应用程序，另一个仅允许访问某些特定的应用程序。有关详细信息，请参阅委派管理文档。
- 使用 SmartAccess 策略表达式过滤通过 NetScaler Gateway 进行的用户连接，从而限制用户的访问权限。
 1. 在 Studio 导航窗格中选择交付组。
 2. 选择交付组，然后在“操作”窗格中选择编辑交付组。
 3. 在访问策略页面上，选择通过 NetScaler Gateway 的连接。
 4. 要选择这些连接的一个子集，请选择满足以下任意过滤器条件的连接。然后定义 NetScaler Gateway 站点，并为允许的用户访问方案添加、编辑或删除 SmartAccess 策略表达式。有关详细信息，请参阅 NetScaler Gateway 文档。
- 通过适用于您在 SDK 中设置的访问策略的排除过滤器限制用户的访问权限。访问策略应用于交付组，以对连接进行细化设置。例如，您可以仅限某个用户子集访问计算机，也可以指定允许的用户设备。排除过滤器可进一步细化访问策略。例如，

出于安全性考虑，您可以拒绝某个用户子集或设备访问。

默认情况下，排除过滤器处于禁用状态。

例如，对于企业网络子网上的教学实验室，要阻止从实验室访问某个特定交付组，而无论该实验室中的计算机使用者为何人，请使用以下命令：`Set-BrokerAccessPolicy -Name VPDesktops_Direct -ExcludedClientIPFilterEnabled $True -`

可以使用星号 (*) 通配符来匹配以相同策略表达式开头的所有标记。例如，如果在一台计算机中添加标记 `VPDesktops_Direct`，在另一台计算机中添加标记 `VPDesktops_Test`，则在 `Set-BrokerAccessPolicy` 脚本中将标记设置为 `VPDesktops_*` 将同时适用于这两台计算机的过滤器。

更新计算机

1. 在 Studio 导航窗格中选择交付组。
2. 选择交付组，然后在“操作”窗格中选择查看计算机。
3. 选择计算机，然后在“操作”窗格中选择更新计算机。
 - 要选择其他主映像，请选择主映像。然后选择快照。展开选定快照将显示关联的主映像。
 - 要应用更改并通知计算机用户，请选择向最终用户发出通知。然后指定：
 - 何时更新主映像：立即还是下次重新启动时。
 - 重新启动分发时间：同时还是以不同时间重新启动所有计算机。
 - 用户是否可以以及何时收到重新启动的通知，外加他们将收到的消息。

应用程序

May 04, 2016
添加应用程序

要将应用程序添加到交付组中，请执行以下操作：

- 1. 在 Studio 导航窗格中选择交付组。
- 2. 选择交付组。
- 3. 在“操作”窗格中选择添加应用程序。

一个列表显示在基于主映像或计算机目录中的模板所创建的计算机上或者 App-V 管理服务器上发现的应用程序。选择要添加到交付组的一个或多个应用程序。

也可以手动添加（创建）应用程序。您需要提供可执行文件路径、工作目录、可选命令行参数以及管理员和用户的显示名称。

您可以更改应用程序的属性；请参阅下文。

默认情况下，您添加的应用程序位于 Applications 文件夹中。有关应用程序文件夹的详细信息，请参阅下文。

复制、禁用、重命名、编辑标记或删除应用程序

要复制、禁用、重命名、编辑标记或删除应用程序，请执行以下操作：

- 1. 在 Studio 导航窗格中选择交付组。
- 2. 在中间窗格中选择“应用程序”选项卡，然后选择应用程序。
- 3. 在“操作”窗格中，选择相应的任务。

须知事项：

- 复制应用程序时，应用程序会自动重命名并置于与原始应用程序相邻的位置。
- 删除某个应用程序会将其从交付组中删除，但并不从主映像中删除。
- 要将应用程序移动到其他应用程序文件夹，请参阅下文。

更改应用程序属性

要更改应用程序的属性，请执行以下操作：

- 1. 在 Studio 导航窗格中选择交付组。
- 2. 在中间窗格中选择“应用程序”选项卡，然后选择应用程序。
- 3. 在“操作”窗格中选择属性。

您可以查看和更改下列项：

要查看或更改的属性	选择此页面
应用程序名称	标识
Receiver 中的类别	交付
命令行参数	位置
说明	标识

要查看或更改的属性 文件扩展名	选择此页面 文件类型关联
文件类型关联	文件类型关联
图标	交付
StoreFront 的关键字	标识
可执行文件夹路径	位置
用户桌面上的快捷方式	交付
可见性	限制可见性
工作目录	位置

在当前的应用程序用户注销其会话之前，应用程序更改可能不对其生效。

管理应用程序文件夹

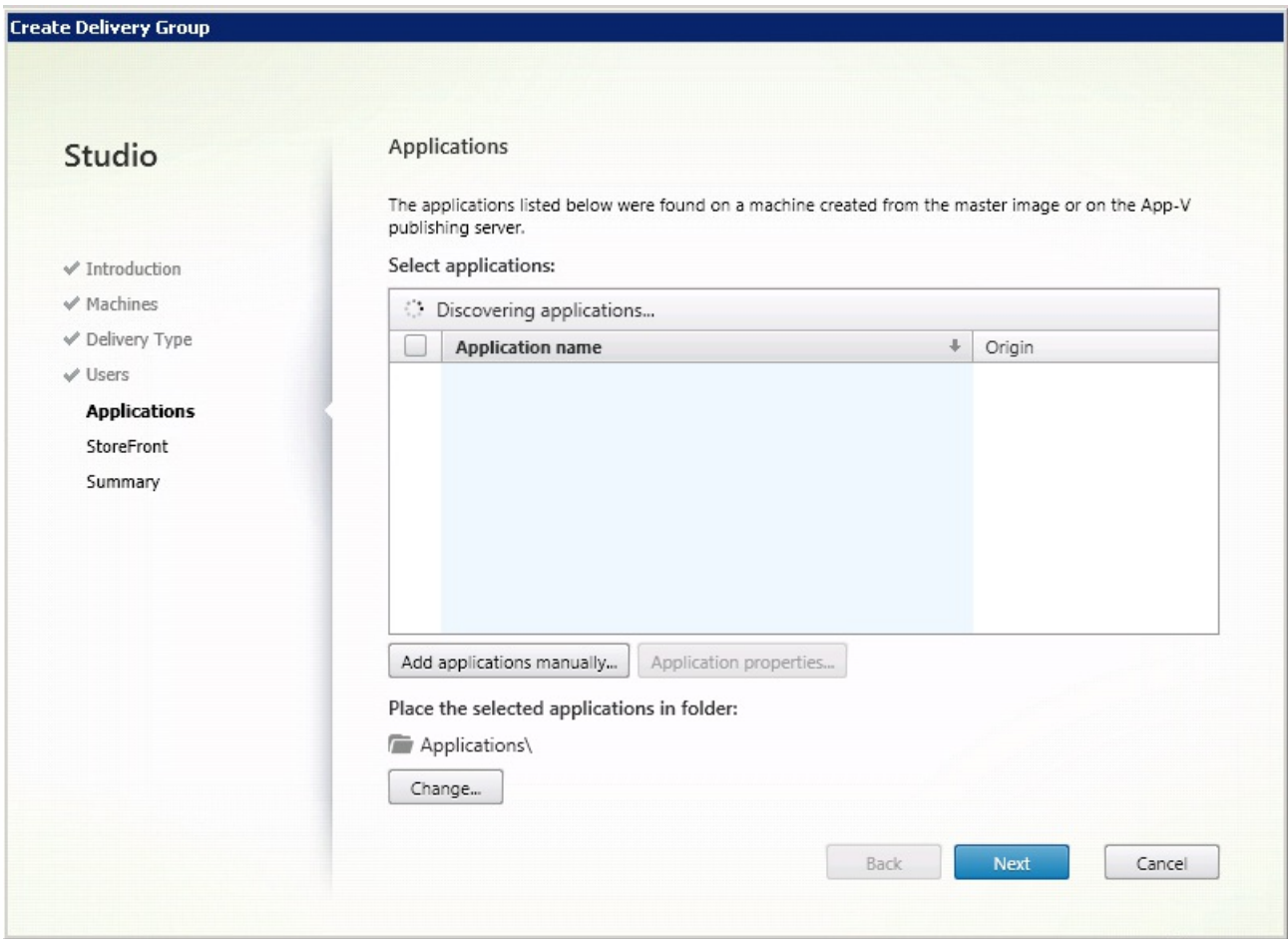
默认情况下，您添加的应用程序位于
— *Applications*
文件夹中。可以执行以下操作：



- 创建其他文件夹，然后将应用程序移到这些新文件夹中。

- 文件夹最多可以嵌套五个级别。
- 文件夹并非必须包含应用程序；它们可以为空。
- 除非您在创建文件夹时对其进行移动或指定其他位置，否则文件夹按字母顺序列出。
- 您可以具有多个名称相同的文件夹，只要其父文件夹不同即可。同样，您可以具有多个名称相同的应用程序，只要其位于不同的文件夹中即可。
- 将文件夹移动到相同或不同级别。最简单的移动方法是拖放。
- 重命名或删除已创建的文件夹。您无法重命名或删除 Applications 文件夹，但可以将其包含的所有应用程序移动到您创建的其他文件夹。

也可以在创建交付组时，为应用程序创建文件夹。

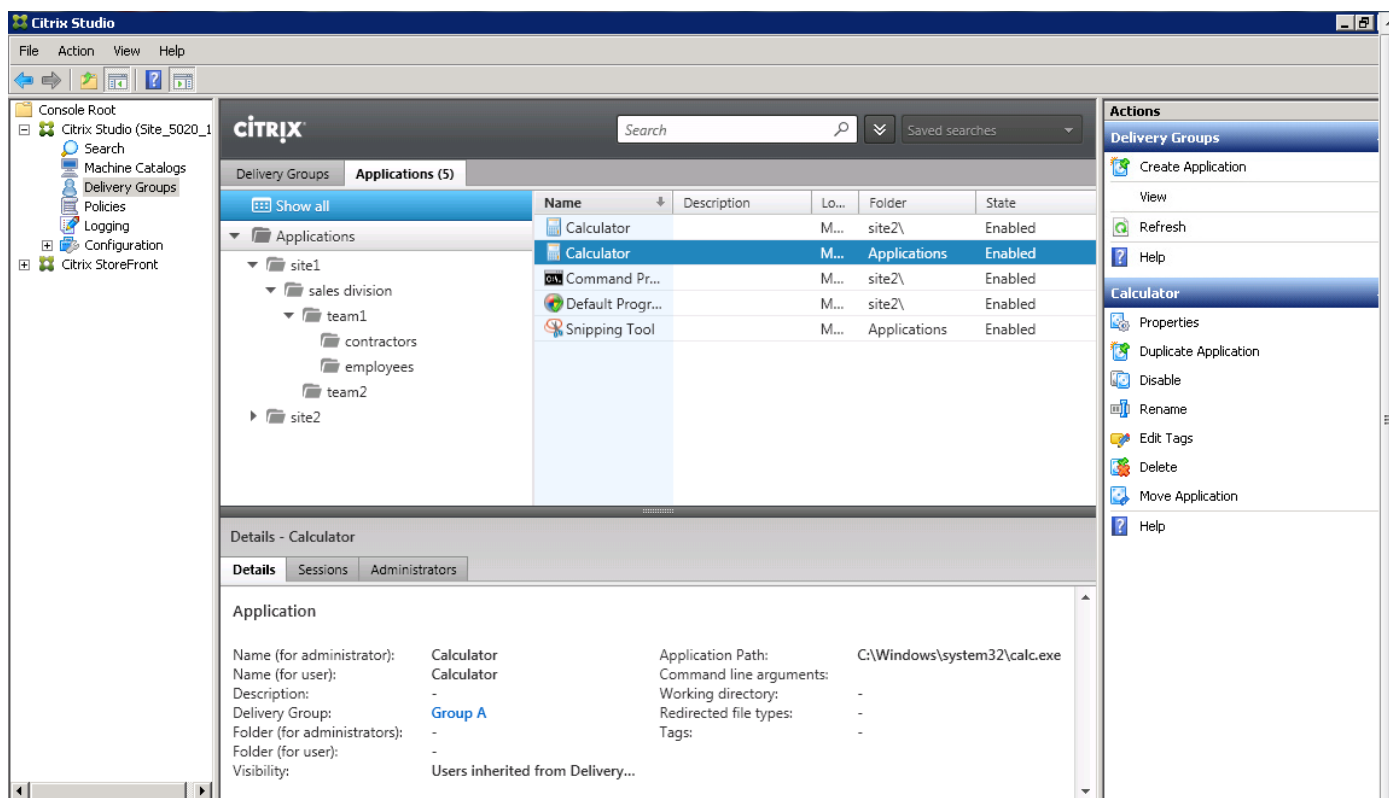


您必须具有查看应用程序权限才能查看文件夹中的应用程序；必须对文件夹中的所有应用程序都具有编辑应用程序属性权限，才能移除、重命名或删除包含应用程序的文件夹。有关详细信息，请参阅[委派管理](#)。

提示：以下说明使用 Studio 中的“操作”窗格。也可以在菜单上单击鼠标右键或拖放。如果您在不理想的位置创建或移动了文件夹，则可以将文件夹拖放到正确位置。

在 Studio 导航窗格中选择交付组，然后在中间窗格中选择应用程序选项卡。

- 要查看所有文件夹（不包括嵌套文件夹），请单击全部显示。



- 创建文件夹：

1. 要将新文件夹置于最高级别（不嵌套在其他文件夹之下），请选择顶部的 Applications 文件夹。要将新文件夹置于 Applications 之外的其他现有文件夹下，请选择该文件夹。

2. 在“操作”窗格中选择创建文件夹。为文件夹输入名称（1-64 个字符）。允许使用空格。

- 要移动文件夹，请执行以下操作：

1. 选择文件夹，然后在“操作”窗格中选择移动文件夹。（一次只能移动一个文件夹，除非文件夹包含嵌套文件夹。）

2. 要将文件夹移动到最高级别（不嵌套在其他文件夹之下），请选择顶部的 Applications 文件夹。要将新文件夹移动到 Applications 之外的其他现有文件夹下，请选择该文件夹。

- 要重命名某个文件夹，请选择该文件夹，然后在“操作”窗格中选择重命名文件夹。输入新名称（1-64 个字符）。

- 要删除某个文件夹，请选择该文件夹，然后在“操作”窗格中选择删除文件夹。删除包含应用程序和其他文件夹的某个文件夹时，这些对象也随之删除。通过删除应用程序，可将分配的应用程序从交付组删除，但不会将其从计算机中删除。

- 要将应用程序移动到文件夹中，请选择一个或多个应用程序，然后在“操作”窗格中选择移动应用程序。选择文件夹。

要在“创建交付组”向导中将应用程序添加或移动到文件夹，请在应用程序页面选择一个或多个应用程序，然后选择更改。

- 要将应用程序移动到现有文件夹，请选择该文件夹。

- 要将应用程序移动到新文件夹，请执行以下操作：

- 要在最高级别创建文件夹（不嵌套在其他文件夹之下），请选择顶部的 Applications 文件夹，然后选择新建文件夹。指定文件夹名称（1-64 个字符）。允许使用空格。

- 要在现有文件夹（非 Applications）下创建新的嵌套文件夹，请选择现有文件夹，然后选择新建文件夹。指定文件夹名称（1-64 个字符）。允许使用空格。

用户

May 04, 2016

添加用户、删除用户以及启用/禁用对未经身份验证（匿名）的用户的访问

用户类型有两种：已通过身份验证和未经身份验证（后者也称为匿名）。您可以配置两种类型之一或全部。

- **已通过身份验证** - 您按名称指定的用户和组成员必须向 StoreFront 或 Citrix Receiver 提供凭据（如智能卡或用户名和密码），才能访问应用程序和桌面。
- **未经身份验证（匿名）** - 对于包含服务器操作系统计算机的交付组，您可以选中特定的复选框，从而允许用户访问应用程序和桌面而不向 StoreFront 或 Citrix Receiver 提供凭据。例如，当用户通过 kiosk 访问应用程序时，应用程序可能需要凭据，但 Citrix 访问门户和工具则不需要。安装 VDA 时，会创建匿名用户组。
 - 要向未经身份验证的用户授予访问权限，交付组中的每台计算机必须已安装 VDA for Windows Server OS（最低版本为 7.6）。启用未经身份验证的用户时，您必须具有未经身份验证的 StoreFront 存储。
 - 启动会话时，将按需创建未经身份验证的用户帐户，其名称为 AnonXYZ，其中，XYZ 是一个唯一的三位数值。
 - 未经身份验证的用户会话的默认空闲超时为 10 分钟，当客户端断开连接时，这些会话将自动注销。不支持重新连接、客户端之间漫游以及 Workspace Control。

1. 在 Studio 导航窗格中选择交付组。
2. 选择交付组，然后在“操作”窗格中选择编辑交付组。
3. 下表介绍了您的选项。

启用访问权限的用户	是否添加/分配用户和用户组？	是否启用“Give access to unauthenticated users”（向未经身份验证的用户授予访问权限）复选框？
仅限已通过身份验证的用户	是	否
仅未经身份验证的用户	否	是
已通过身份验证的用户和未经身份验证的用户	是	是

对于包含桌面操作系统计算机的桌面组，您可以在创建交付组之后导入用户数据（用户列表）。请参阅下面的[导入或导出用户列表](#)。

导入或导出用户列表

对于包含物理桌面操作系统计算机的交付组，您可以在创建交付组之后从 .csv 文件导入用户信息。您还可以将用户信息导出到 .csv 文件。.csv 文件可以包含来自先前产品版本的数据。

.csv 文件中的第一行必须包含以逗号分隔的列标题（任何顺序均可），其中可以包括：ADComputerAccount、AssignedUser、VirtualMachine 和 HostId。文件中的后续行包含以逗号分隔的数据。ADComputerAccount 条目可以是公用名、IP 地址、标识名或域和计算机名称对。

要导入或导出用户信息，请执行以下操作：

1. 在 Studio 导航窗格中选择交付组。
2. 选择交付组，然后在“操作”窗格中选择编辑交付组。
3. 在计算机分配页面上，选择导入列表或导出列表按钮，然后浏览到文件位置。

会话

Nov 02, 2016

Log off or disconnect a session, or send a message to users

1. Select Delivery Groups in the Studio navigation pane.
2. Select a Delivery Group and then select View Machines in the Actions pane.
3. To log a user off a session, select the session or desktop and select Log off in the Actions pane. The session closes and the machine becomes available to other users, unless it is allocated to a specific user.
To disconnect a session, select the session or desktop, and select Disconnect in the Actions pane. Applications continue to run and the machine remains allocated to that user. The user can reconnect to the same machine.

To send a message to users, select the session, machine, or user, and then select Send message in the Actions pane. Enter the message.

You can configure power state timers for Desktop OS machines to automatically handle unused sessions. See [Power manage machines](#) for details.

Configure session prelaunch and session linger

Note: These features are supported on Server OS machines only.

The session prelaunch and session linger features help specified users access applications quickly, by starting sessions before they are requested (session prelaunch) and keeping application sessions active after a user closes all applications (session linger).

By default, session prelaunch and session linger are not used: a session starts (launches) when a user starts an application, and remains active until the last open application in the session closes.

Considerations:

- The Delivery Group must support applications, and the machines must be running a VDA for Server OS, minimum version 7.6.
- Session prelaunch is supported only when using Citrix Receiver for Windows. Session linger is supported when using Citrix Receiver for Windows and Receiver for Web. Additional Receiver configuration is required. For instructions, search for "session prelaunch" in the eDocs content for your Receiver for Windows version.
Note: Receiver for HTML5 is not supported.
- When using session prelaunch:
 - Regardless of the admin-side settings, if an end user's machine is put into "suspend" or "hibernate" mode, prelaunch

will not work.

- Prelaunch will work as long as the end user locks their machine/session, but if the end user logs off from Citrix Receiver, the session is ended and prelaunch no longer applies.
- Prelaunched and lingering sessions consume a license, but only when connected. Unused prelaunched and lingering sessions disconnect after 15 minutes by default. This value can be configured in PowerShell (New/Set-BrokerSessionPreLaunch cmdlet).
- Careful planning and monitoring of your users' activity patterns are essential to tailoring these features to complement each other. Optimal configuration balances the benefits of earlier application availability for users against the cost of keeping licenses in use and resources allocated.
- You can also configure session prelaunch for a scheduled time of day in Receiver.

To enable session prelaunch:

1. Select Delivery Groups in the Studio navigation pane.
2. Select a Delivery Group, and then click Edit Delivery Group in the Actions pane.
3. On the Application Prelaunch page, enable session prelaunch by choosing when sessions should launch:
 - When a user starts an application. This is the default setting; session prelaunch is disabled.
 - When any user in the Delivery Group logs on to Receiver for Windows.
 - When anyone in a list of users and user groups logs on to Receiver for Windows. Be sure to also specify users or user groups if you choose this option.

Edit Delivery Group

Studio

- Users
- Delivery Type
- Application Prelaunch**
- Application Linging
- Basic settings
- Access Policy
- Restart Schedule

Prelaunch Sessions for Applications

With prelaunch, sessions launch when users log on to Receiver, so applications are available sooner.

When do you want sessions to launch?

- ☐ Launch when users start an application (no prelaunch)
- ☒ Prelaunch when any user in the Delivery Group logs on to Receiver for Windows
- ☐ Prelaunch when any of the following users log on to Receiver for Windows:

Add... Remove

If no application is started, when do you want prelaunched sessions to end?

After a specified time: Hours 2 - +

☒ When average load on all machines exceeds (%): 80 - +

☒ When load on any machine exceeds (%): 85 - +

OK Cancel Apply

4. A prelaunched session is replaced with a regular session when the user starts an application. If the user does not start an application (the prelaunched session is unused), the following settings affect how long that session remains active. For

details about these settings, see

— *How long unused prelaunched and lingering sessions remain active*

below.

- When a specified time interval elapses. You can change the time interval (1-99 days, 1-2376 hours, or 1-142,560 minutes).
- When the average load on all machines in the Delivery Group exceeds a specified percentage (1-99%).
- When the load on any machine in the Delivery Group exceeds a specified percentage (1-99%).

Recap: A prelaunched session remains active until one of the following events occurs: a user starts an application, the specified time elapses, or a specified load threshold is exceeded.

To enable session linger:

1. Select Delivery Groups in the Studio navigation pane.
2. Select a Delivery Group, and then click Edit Delivery Group in the Actions pane.
3. On the Application Linging page, enable session linger by selecting the Keep sessions active until radio button.

Edit Delivery Group

Studio

- Users
- Delivery Type
- Application Prelaunch
- Application Linging**
- Basic settings
- Access Policy
- Restart Schedule

Lingering Sessions for Applications

With lingering, sessions remain active after all applications are closed.

When do you want sessions to end?

☐ Immediately after all applications in the session are closed (no lingering)

☒ Keep sessions active until:

After a specified time:

Hours 8 - +

☒ The average load on all machines exceeds (%):

70 - +

☒ The load on any machine exceeds (%):

70 - +

OK Cancel Apply

4. Several settings affect how long a lingering session remains active if the user does not start another application. For details about these settings, see

— *How long prelaunched and lingering sessions remain active*

below.

- When a specified time interval elapses. You can change the time interval (1-99 days, 1-2376 hours, or 1-142,560 minutes).
- When the average load on all machines in the Delivery Group exceeds a specified percentage (1-99%).
- When the load on any machine in the Delivery Group exceeds a specified percentage (1-99%).

Recap: A lingering session remains active until one of the following events occurs: a user starts an application, the

specified time elapses, or a specified load threshold is exceeded.

How long unused prelaunched and lingering sessions remain active - There are several ways to specify how long an unused session remains active if the user does not start an application: a configured timeout and server load thresholds. You can configure all of them; the event that occurs first will cause the unused session to end.

- **Timeout** - A configured timeout specifies the number of minutes, hours, or days an unused prelaunched or lingering session remains active. If you configure too short a timeout, prelaunched sessions will end before they provide the user benefit of quicker application access. If you configure too long a timeout, incoming user connections might be denied because the server doesn't have enough resources.

You cannot disable this timeout from Studio, but you can in the SDK (`New/Set-BrokerSessionPreLaunch` cmdlet). If you disable the timeout, it will not appear in the Studio display for that Delivery Group or in the Edit Delivery Group wizard.

- **Thresholds** - Automatically ending prelaunched and lingering sessions based on server load ensures that sessions remain open as long as possible, assuming server resources are available. Unused prelaunched and lingering sessions will not cause denied connections because they will be ended automatically when resources are needed for new user sessions. You can configure two thresholds: the average percentage load of all servers in the Delivery Group, and the maximum percentage load of a single server in the Delivery Group. When a threshold is exceeded, the sessions that have been in the prelaunch or lingering state for the longest time are ended, sessions are ended one-by-one at minute intervals until the load falls below the threshold. (While the threshold is exceeded, no new prelaunch sessions are started.)

Servers with VDAs that have not registered with the Controller, and servers in maintenance mode are considered fully loaded. An unplanned outage will cause prelaunch and lingering sessions to be ended automatically to free capacity.

XenApp 发布的应用程序和桌面

May 04, 2016

使用服务器操作系统计算机交付 XenApp 发布的应用程序和 XenApp 发布的桌面。

此表格描述了使用这些交付方法的情况、用户和注意事项。

用例	您希望 基于服务器的经济实惠的交付，以便最大程度地减少向大量用户交付应用程序的成本，同时提供安全的高清晰度用户体验。 您的用户 执行定义明确的任务且不需要个性化设置或应用程序脱机访问权限。用户可能包括任务型工作人员（如呼叫中心操作人员和零售工作人员）或共享工作站的用户。 应用程序类型 任何应用程序。
优势和注意事项	优势 数据中心内可管理、可扩展的解决方案。 最经济的应用程序交付解决方案。 托管应用程序集中进行管理且用户无法修改应用程序，从而提供一致、安全、可靠的用户体验。 注意事项 用户必须联机才能访问其应用程序。
用户体验	用户可通过 StoreFront、“开始”菜单或您为其提供的 URL 请求一个或多个应用程序。 应用程序以虚拟方式进行交付并在用户设备上高清晰度无缝显示。 根据配置文件设置，用户所做的更改会在用户的应用程序会话结束时进行保存。否则，这些更改将被删除。
处理、托管和交付应用程序	进程 应用程序处理在托管计算机（而非用户设备）上执行。 托管计算机可以是物理机，也可以是虚拟机。 主机 应用程序和桌面驻留在服务器操作系统计算机上。 计算机通过计算机目录提供。

	交付 计算机目录中的计算机分为可将一组相同的应用程序交付给用户组的交付组。 服务器操作系统计算机支持： • 同时托管桌面和应用程序的桌面和应用程序交付组。 • 仅托管应用程序的应用程序交付组。
会话管理 和分配	会话 服务器操作系统计算机可从单台计算机运行多个会话，以便将多个应用程序和桌面交付给多个同时连接的用户。每个用户均需要可从中运行其所有托管应用程序的单个会话。 例如，一个用户登录并请求某个应用程序。该计算机上的一个会话变为对其他用户不可用。另一个用户登录并请求该计算机托管的应用程序。同一台计算机上的另一个会话现在不可用。如果两个用户同时请求其他应用程序，则不需要任何其他会话，因为用户可以使用同一个会话运行多个应用程序。如果有另外两个用户登录并请求桌面且同一台计算机上存在两个可用会话，该计算机现在将使用四个会话托管四个不同的用户。 随机计算机分配 在分配有用户的交付组内，将选择负载最低的服务器上的计算机。具有可用会话的计算机将随机分配，用以在用户登录时向用户交付应用程序。

要交付 XenApp 发布的应用程序，请执行以下操作：

1. 在运行受支持的 Windows 服务器操作系统的主映像上安装要交付的应用程序。
2. 创建此主映像的计算机目录或使用主映像更新现有目录。
3. 创建应用程序交付组以向用户交付应用程序。
4. 从安装的应用程序列表中，选择要交付的应用程序。

要交付 XenApp 发布的桌面，请执行以下操作：

1. 在运行受支持的 Windows 服务器操作系统的主映像上安装应用程序。
2. 创建此主映像的计算机目录或使用主映像更新现有目录。
3. 创建桌面交付组以向用户交付桌面。

VM 托管应用程序

May 04, 2016

使用桌面操作系统计算机交付 VM 托管应用程序。

此表格描述了适合使用此交付方法的情况、用户和注意事项。

用例	您希望 基于客户端的安全应用程序交付解决方案可提供集中管理功能，并支持每台主机服务器（或虚拟机管理程序）具有大量用户，同时为用户提供以高清晰度无缝显示的应用程序。 您的用户 为内外部承包商、第三方合作者及其他临时团队成员。 用户不需要脱机访问托管应用程序。 应用程序类型 可能不会与其他应用程序完美配合使用或可能与操作系统进行交互的应用程序，例如 Microsoft .NET Framework。这些类型的应用程序最适合在虚拟机上进行托管。 在旧版操作系统（如 Windows XP 或 Windows Vista）或旧版体系结构（32 位或 16 位）上运行的应用程序。通过将每个应用程序隔离在其各自的虚拟机上，如果一台计算机发生故障，也不会影响其他用户。
优势和注意事项	优势 可在数据中心内的计算机上安全管理、托管和运行主映像上的应用程序和桌面，从而提供一个更为经济的应用程序交付解决方案。 • 登录后，可以将用户随机分配给交付组内配置为托管相同应用程序的计算机。 • 还可以静态分配单台计算机，以便在每次有单个用户登录时将应用程序交付给该用户。通过静态分配的计算机，用户可以在虚拟机上安装和管理自己的应用程序。 注意事项 桌面操作系统计算机上不支持运行多个会话。因此，登录后，每个用户都将占用交付组内的单台计算机，且这些用户必须联机才能访问其应用程序。 此方法可能会增加用于处理应用程序的服务器资源量，同时增加用户的个人虚拟磁盘的存储量。
用户体验	与在服务器操作系统计算机上托管共享应用程序相同的无缝应用程序体验。
处理、托管和交付应用程序	进程 与服务器操作系统计算机相同，但这些计算机属于虚拟桌面操作系统计算机。 主机

	与服务器操作系统计算机相同，但这些计算机属于虚拟桌面操作系统计算机。 交付 与服务器操作系统计算机相同，但桌面操作系统计算机仅可存在于桌面交付组中。
会话管理和分配	会话 桌面操作系统计算机可从单台计算机运行单个桌面会话。仅当访问应用程序时，单个用户才能使用多个应用程序（不限于单个应用程序），因为操作系统将每个应用程序视为一个新会话。 随机和静态计算机分配 在分配有用户的交付组内，用户在登录后即可访问： • 静态分配的计算机，以使用户每次都可登录到同一台计算机。 • 基于会话可用性选择的随机分配的计算机。

要交付 VM 托管应用程序，请执行以下操作：

1. 在运行受支持的 Windows 桌面操作系统的主映像上安装要交付的应用程序。
2. 创建此主映像的计算机目录或使用主映像更新现有目录。
定义计算机目录的桌面体验时，确定用户每次登录时是连接到新 VM 还是连接到相同的 VM。
3. 创建应用程序交付组以向用户交付应用程序。
4. 从安装的应用程序列表中，选择要交付的应用程序。

VDI 桌面

May 04, 2016

使用桌面操作系统计算机交付 VDI 桌面。

VDI 桌面托管在虚拟机上，并向每个用户提供桌面操作系统。

VDI 桌面需要的资源高于 XenApp 发布的桌面，但是不要求其上安装的应用程序支持基于服务器的操作系统。此外，根据您选择的 VDI 桌面类型，可以将这些桌面分配给单个用户，并允许这些用户进行高度个性化设置。

创建 VDI 桌面的计算机目录时，创建以下桌面类型之一：

- 随机非永久性桌面，也称为池 VDI 桌面。用户每次登录以使用其中一个桌面时，将连接到从基于单个主映像的桌面池中动态选择的桌面。计算机重新启动时，对桌面所做的更改将全部丢失。
- 静态非永久性桌面。用户首次登录以使用其中一个桌面时，将向此用户分配基于单个主映像的桌面池中的桌面。第一次使用后，用户每次登录以使用其中的桌面时，将连接到此用户第一次使用时向其分配的桌面。计算机重新启动时，对桌面所做的更改将全部丢失。
- 静态永久性桌面，也称为使用个人虚拟磁盘的 VDI。与其他 VDI 桌面类型不同，用户完全可以对这些桌面进行个性化设置。用户首次登录以使用其中一个桌面时，将向此用户分配基于单个主映像的桌面池中的桌面。第一次使用后，用户每次登录以使用其中的桌面时，将连接到此用户第一次使用时向其分配的桌面。计算机重新启动时，会保留对桌面所做的更改，因为这些更改存储在个人虚拟磁盘中。

要交付 VDI 桌面，请执行以下操作：

1. 创建运行受支持的 Windows 桌面操作系统的主映像。
2. 创建此主映像的计算机目录或使用主映像更新现有目录。
定义有关计算机目录的桌面体验时，确定用户在每次登录时连接到新的 VM 还是在每次登录时连接到同一个计算机，并指定如何保留对桌面所做的更改。
3. 创建桌面交付组以向用户交付桌面。

Remote PC Access

Nov 02, 2016

Remote PC Access allows an end user to log on remotely from virtually anywhere to the physical Windows PC in the office. The Virtual Delivery Agent (VDA) is installed on the office PC; it registers with the Delivery Controller and manages the HDX connection between the PC and the end user client devices. Remote PC Access supports a self-service model; after you set up the whitelist of machines that users are permitted to access, those users can join their office PCs to a Site themselves, without administrator intervention. The Citrix Receiver running on their client device enables access to the applications and data on the office PC from the Remote PC Access desktop session.

A user can have multiple desktops, including more than one physical PC or a combination of physical PCs and virtual desktops.

Note: Sleep mode & Hibernation mode for Remote PC is not supported. Remote PC Access is valid only for XenDesktop licenses; sessions consume licenses in the same way as other XenDesktop sessions.

Active Directory considerations:

- Before configuring the remote PC deployment site, set up your Organizational Units (OUs) and security groups and then create user accounts. Use these accounts to specify users for the Delivery Groups you will use to provide Remote PC Access.
- If you modify Active Directory after a machine has been added to a machine catalog, Remote PC Access does not reevaluate that assignment. You can manually reassign a machine to a different catalog, if needed.
- If you move or delete OUs, those used for Remote PC Access can become out of date. VDAs might no longer be associated with the most appropriate (or any) machine catalog or Delivery Group.

Machine catalog and Delivery Group considerations:

- A machine can be assigned to only one machine catalog and one Delivery Group at a time.
- You can put machines in one or more Remote PC Access machine catalogs.
- When choosing Machine Accounts for a machine catalog, select the lowest applicable OU to avoid potential conflicts with machines in another catalog. For example, in the case of Bank/officers/tellers, select tellers.
- You can allocate all machines from one remote PC machine catalog through one or more Delivery Groups. For example, if one group of users requires certain policy settings and another group requires different settings, assigning the users to different Delivery Groups enables you to filter the HDX policies according to each Delivery Group.
- If your IT infrastructure assigns responsibility for servicing users based on geographic location, department, or some other category, you can group machines and users accordingly to allow for delegated administration. Ensure that each administrator has permissions for both the relevant machine catalogs and the corresponding Delivery Groups.
- For users with office PCs running Windows XP, create a separate machine catalog and Delivery Group for those systems. When choosing machine accounts for that catalog in Studio, select the checkbox indicating that some machines are running Windows XP.

Deployment considerations:

- You can create a Remote PC Access deployment and then add traditional Virtual Desktop Infrastructure (VDI) desktops or applications later. You can also add Remote PC Access desktops to an existing VDI deployment.
- Consider whether to enable the Windows Remote Assistance feature when you install the VDA on the office PC. This option allows help desk teams using Director to view and interact with a user sessions using Windows Remote Assistance.
- Consider how you will deploy the VDA to each office PC. Citrix recommends using electronic software distribution such as Active Directory scripts and Microsoft System Center Configuration Manager. The installation media contains sample

Active Directory scripts.

- Secure Boot functionality is currently unsupported. Disable Secure Boot if intending to deploy the workstation VDA.
- Each office PC must be domain-joined with a wired network connection.
- Windows 7 Aero is supported on the office PC, but not required.
- Connect the keyboard and mouse directly to the PC or laptop, not to the monitor or other components that can be turned off. (If you must connect input devices to components such as monitors, they should not be turned off.)
- If you are using smart cards, see [Smart cards](#).
- Remote PC Access can be used on most laptop computers. To improve accessibility and deliver the best connection experience, configure the laptop power saving options to those of a desktop PC. For example:
 - Disable the Hibernate feature.
 - Disable the Sleep feature.
 - Set the close lid action to Do Nothing.
 - Set the press the power button action to Shut Down.
 - Disable video card energy saving features.
 - Disable network interface card energy saving features.
 - Disable battery saving technologies.

The following are not supported for Remote PC Access devices:

- Docking and undocking the laptop.
- KVM switches or other components that can disconnect a session.
- Hybrid PCs (including All-in-One and NVIDIA Optimus laptops and PCs) and Surface Pro/Books.
- Install Citrix Receiver on each client device that remotely accesses the office PC.
- Multiple users with remote access to the same office PC see the same icon in Receiver. When any user remotely logs on to the PC, that resource appears as unavailable to other users.
- By default, a remote user's session is automatically disconnected when a local user initiates a session on that machine (by pressing CTRL+ALT+DEL). To prevent this automatic action, add the following registry entry on the office PC, and then restart the machine.

Caution: Editing the registry incorrectly can cause serious problems that may require you to reinstall your operating system. Citrix cannot guarantee that problems resulting from the incorrect use of Registry Editor can be solved. Use Registry Editor at your own risk. Be sure to back up the registry before you edit it.

HKLM\SOFTWARE\Citrix\PortICA\RemotePC "SasNotification"=dword:00000001

To further customize the behavior of this feature under HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\PortICA\RemotePC

- RpcMode (dword)
- RpcTimeout (dword)

RpcMode:

1 - Means that the remote user will always win if he does not respond to the Messaging UI in the specified timeout period.

2 - Means that the Local user will always win. If this setting is not specified, the Remote user will always win by default.

RpcTimeout:

The number of seconds given to the user before we automatically decide which type of mode to enforce. If this setting is not specified, the default value is :30 seconds. The minimum value here should be :30 seconds. The User needs to restart the machine for these changes to take place.

When user wants to forcibly get the console access: The local user can hit Ctr+Alt+Del twice in a gap of :10 seconds to get local control over a remote session and force a disconnect event.

After the registry change and machine restart, if a local user presses CTRL+ALT+DEL to log on to that PC while it is in use by a remote user, the remote user receives a prompt asking whether or not to allow or deny the local user's connection. Allowing the connection will disconnect the remote user's session.

The following XenDesktop features are not supported for Remote PC Access deployments:

- Creating master images and virtual machines
- Delivering hosted applications
- Personal vDisks
- Client folder redirection

Wake on LAN

Remote PC Access supports Wake on LAN, which gives users the ability to turn on physical PCs remotely. This feature enables users to keep their office PCs turned off when not in use, saving energy costs. It also enables remote access when a machine has been turned off inadvertently, such as during weather events.

With XenDesktop 7.6 Feature Pack 3, Citrix released an experimental Wake on LAN SDK. This enables you or a third-party Wake on LAN solution to create a connector without the requirement of System Center 2012 R2. For more information, see <http://support.citrix.com/article/CTX202272>.

The Remote PC Access Wake on LAN feature is supported on both of the following:

- PCs that support Intel Active Management Technology (AMT)
- PCs that have the Wake on LAN option enabled in the BIOS

You must configure Microsoft System Center Configuration Manager (ConfigMgr) 2012 to use the Wake on LAN feature. ConfigMgr provides access to invoke AMT power commands for the PC, plus Wake-up proxy and magic-packet support. Then, when you use Studio to create a Remote PC Access deployment (or when you add another power management connection to be used for Remote PC Access), you enable power management and specify ConfigMgr access information.

Additionally:

- Using AMT power operations is preferred for security and reliability; however, support is also provided for two non-AMT methods: ConfigMgr Wake-up proxy and raw magic packets.
- On AMT-capable machines only, the Wake on LAN feature also supports the Force-Shutdown and Force-Restart actions in Studio and Director. Additionally, a Restart action is available in StoreFront and Receiver.

For more information, see [Configuration Manager and Remote PC Access Wake on LAN](#) and [Provide users with Remote PC Access](#).

向用户提供 Remote PC Access

May 04, 2016

借助 Remote PC Access，桌面用户可以安全访问 办公室 PC 上的资源，同时体验 Citrix HDX 技术的优势。

注意：Remote PC Access 仅对 XenDesktop 许可证有效。

1. 要使用 Remote PC Access 电源管理功能（又称 Remote PC Access 局域网唤醒），在 Studio 中创建 Remote PC Access 部署前，请在 PC 和 Microsoft System Center Configuration Manager (ConfigMgr) 2012 上完成配置任务。有关详细信息，请参阅 [Configuration Manager](#) 和 [Remote PC Access 局域网唤醒](#)。
2. 创建初始 Remote PC Access 部署时，可以为默认 Remote PC Access 计算机目录中的计算机启用或禁用电源管理。如果启用电源管理，请指定 ConfigMgr 连接信息。然后指定用户和计算机帐户。有关详细信息，请参阅[创建站点](#)。创建 Remote PC 部署不会阻止将来在此站点中使用 VDI。
创建 Remote PC Access 部署时会同时创建一个名为
— *Remote PC Access Machines*
的默认计算机目录和一个名为
— *Remote PC Access Desktops*
的默认交付组。
3. 创建用于 Remote PC Access 的另一个计算机目录时：
 - 操作系统：选择 Remote PC Access，然后选择一个电源管理连接。也可以选择不使用电源管理。如果尚未配置电源管理连接，可以在完成计算机目录创建向导后添加一个连接（连接类型 = Microsoft Configuration Manager 局域网唤醒），然后编辑计算机目录，指定此新连接。
 - 计算机帐户：可以从计算机帐户或显示的组织单位 (OU) 中进行选择，也可以添加计算机帐户和 OU。
4. 在用于本地和远程访问的办公室 PC 上安装 VDA。通常情况下，可通过使用软件包管理软件自动部署 VDA；但是，对于概念验证或小型部署，可以在每台办公室 PC 上手动安装 VDA。
安装 VDA 后，登录办公室 PC 上的控制台会话（在本地或通过 RDP）的下一个域用户将自动分配给 Remote PC 桌面。如果其他域用户登录控制台会话，这些用户也将添加到桌面用户列表，但需符合用户配置的限制。
注意：要在 XenApp 或 XenDesktop 环境外部使用 RDP 连接，必须将用户或组添加到直接访问用户组。
5. 指示用户在用于远程访问办公室 PC 的每台客户端设备上下载并安装 Citrix Receiver。可以从 <http://www.citrix.com> 或支持的移动设备的应用程序分发系统获取 Citrix Receiver。

可以编辑电源管理连接以配置高级设置。可以启用：

- ConfigMgr 提供的唤醒代理。
- 局域网唤醒（幻）数据包。如果启用局域网唤醒数据包，则可以选择局域网唤醒传输方法：“子网定向广播”或“单播”。

PC 使用 AMT 电源命令（如果支持）以及启用的任何高级设置。如果 PC 不使用 AMT 电源命令，则会使用高级设置。

故障排除

Delivery Controller 可将以下 Remote PC Access 诊断信息写入 Windows 应用程序事件日志。信息性消息不受限制。错误消息受限制，需删除重复消息。

- 3300（信息性消息）- 计算机已添加到目录
- 3301（信息性消息）- 计算机已添加到交付组
- 3302（信息性消息）- 计算机已分配给用户
- 3303（错误消息）- 异常

启用 Remote PC Access 电源管理时，子网定向广播可能无法启动与 Controller 不在同一子网上的计算机。如果需要使用子网定向广播跨子网管理电源且不支持 AMT，请尝试使用唤醒代理或“单播”方法（确保在电源管理连接的高级属性中启用这些设置）。

管理 Remote PC Access 交付组

May 04, 2016

如果 Remote PC Access 计算机目录中的计算机未分配给用户，Studio 会暂时将该计算机分配给与此计算机目录关联的交付组。此临时分配会提供信息，以便以后将该计算机分配给用户。交付组与计算机目录的关联具有一个优先级值。

优先级决定向系统注册计算机或用户需要计算机分配时，将计算机分配给哪个交付组。值越小，优先级越高。如果 Remote PC Access 计算机目录具有多个交付组分配，该软件将选择优先级最高的匹配项。您可以使用 PowerShell SDK 设置此优先级值。

添加或删除 Remote PC Access 计算机目录关联

首次创建后，Remote PC Access 计算机目录将与交付组关联。这意味着添加到计算机目录的计算机帐户或组织单位稍后可以添加到交付组。此关联可以关闭或开启。

1. 在 Studio 导航窗格中选择交付组。
2. 选择 Remote PC Access 交付组。
3. 在“详细信息”部分中，选择“目录”选项卡，然后选择 Remote PC Access 计算机目录。
4. 要添加或还原关联，请选择 添加桌面。要删除关联，请选择删除关联。

App-V

May 04, 2016

利用 Microsoft Application Virtualization (App-V)，您可以将应用程序作为服务进行部署、更新及提供支持。这些应用程序无需安装在用户设备上即可供访问。借助 App-V 和 Microsoft User State Virtualization (USV)，用户无论身处何处，是否连接 Internet，均可对应用程序和数据进行访问。

下表列出了支持的版本。（不再支持 App-V 4.6 2 客户端。）

App-V	XenDesktop 和 XenApp 版本	
	Delivery Controller	VDA
5.0	XenDesktop 7 至当前版本 XenApp 7.5 至当前版本	7.0 至当前版本
5.0 SP1	XenDesktop 7 至当前版本 XenApp 7.5 至当前版本	7.0 至当前版本
5.0 SP2	XenDesktop 7 至当前版本 XenApp 7.5 至当前版本	7.1 至当前版本
5.0 SP3 和 5.1	XenDesktop 7.6 XenApp 7.6	7.6.300

受支持的 App-V 客户端不支持脱机访问应用程序。App-V 集成支持包括针对应用程序使用 SMB 共享；不支持 HTTP 协议。

- 无需对操作系统设置进行任何预先配置或更改，即可无缝使用应用程序。App-V 包含下列组件：
- 管理服务器 — 提供一个中央控制台，用于管理 App-V 基础结构，并将虚拟应用程序同时交付给 App-V 桌面客户端和远程桌面服务客户端。App-V 管理服务器将进行身份验证、发出请求并提供管理员所需的安全性、计量、监视及数据收集功能。服务器使用 Active Directory 和支持工具来管理用户和应用程序。
 - 发布服务器 — 为特定用户提供 App-V Client 和应用程序，并托管虚拟应用程序包以进行流技术推送。它从管理服务器提取应用程序包。
 - Client — 检索虚拟应用程序、在客户端发布应用程序，并在运行时自动设置和管理 Windows 设备上的虚拟环境。App-V Client 安装在 VDA 上，且存储特定于用户的虚拟应用程序设置，例如，各个用户配置文件中的注册表和文件更改。

可以从服务器操作系统和桌面操作系统交付组启动 App-V 应用程序：

- 通过 Citrix Receiver
- 从“开始”菜单
- 通过 App-V Client 和 Citrix Receiver
- 同时由多个用户多台设备上启动
- 通过 Citrix StoreFront

应用程序启动时，会实现修改的 App-V 应用程序属性。例如，对于修改过显示名称或具有自定义图标的应用程序，用户启动应用程序时会显示修改内容。

将桌面和应用程序交付组更改为仅应用程序交付组时，App-V 应用程序的性能不会发生变化。

仅支持基于 App-V 服务器的部署，在此部署中，管理员使用 App-V 管理服务器和发布服务器来管理 App-V 应用程序。

配置 App-V

要交付 App-V 应用程序，请执行以下操作：

- 按照 <http://technet.microsoft.com/en-us/virtualization/hh710199> 中所述的说明部署 App-V。
- 在 App-V 管理服务器上发布 App-V 应用程序。配置权限和文件类型关联等设置。如果已部署 App-V，则这些设置均已存在。
- 也可以更改 App-V 发布服务器设置；请参阅下文。
- 在 VDA 上安装 App-V Client。
- 在 Studio 中创建站点期间，指定 App-V 发布服务器和管理服务器的 URL 和端口号。这些服务器自动供交付组使用。
- 在计算机目录的主映像中安装 App-V Client。使用 ShareContentStoreMode 和 EnablePackageScripts 等设置配置该客户端。（不需要在主映像中配置 App-V 发布服务器，因为已在应用程序启动过程中配置。）
- 创建交付组期间，选择 App-V 应用程序。

这些应用程序现在可以使用。

可以在创建站点后指定或更改 App-V 服务器信息。在 Studio 导航窗格中选择配置 > App-V Publishing (App-V 发布)，然后在“操作”窗格中选择相应条目。可以通过为 App-V 管理服务器和发布服务器指定 URL 和端口号来添加 App-V 发布。也可以编辑或删除这些地址。如果刷新 App-V 应用程序，显示内容将指示服务器连接是否存在问题，并且将不再可用的应用程序条目删除。

App-V 发布服务器设置

要更改发布服务器设置，Citrix 建议在 Controller 上使用 SDK cmdlet。

- 要查看发布服务器设置，请输入 Get-CtxAppvServerSetting -AppVPublishingServer。
- 要确保 App-V 应用程序正常启动，请输入 Set-CtxAppvServerSetting -UserRefreshOnLogon 0。

以下 cmdlet 将更改 Controller 上 App-V 发布服务器的设置。并非所有参数都具有强制性。

Set-CtxAppvServerSetting -AppVPublishingServer -UserRefreshOnLogon -UserRefreshEnabled -UserRefreshInterval -UserRefreshIntervalUnit -GlobalRefreshOnLogon -GlobalRefreshEnabled -GlobalRefreshInterval -GlobalRefreshIntervalUnit

注意：如果您以前曾使用 GPO 策略设置管理发布服务器设置，GPO 设置会覆盖任何 App-V 集成设置，包括以前的 cmdlet 设置。这可能会导致 App-V 应用程序启动失败。Citrix 建议您删除所有 GPO 策略设置并使用 SDK 配置相同的设置。

故障排除

- 如果您在 Studio 中指定 App-V 管理服务器和发布服务器地址时，测试连接操作失败，请检查以下各项：
 - App-V 服务器已启动：发送 Ping 命令或检查 IIS 管理器（每个 App-V 服务器都应该处于“已启动”或“正在运行”状态）。
 - 在 App-V 服务器上启用 PowerShell 远程处理。如果未启用，请按照 <http://technet.microsoft.com/en-us/magazine/ff700227.aspx> 中的过程操作。
 - 将 App-V 服务器添加到 Active Directory。

如果 Studio 计算机与 App-V 服务器位于不存在信任关系的不同 Active Directory 域中，请从 Studio 计算机上的 PowerShell 控制台运行 `winrm s winrm/Config/client '@(TrustedHosts="<App-V 服务器 FQDN>")`。如果 TrustedHosts 由 GPO 管理，将显示以下错误消息：“无法更改配置设置 TrustedHosts，因为使用受策略控制。策略需要设置为“未配置”才能更改配置设置”。如果显示此消息，请在 GPO 中向 TrustedHosts 策略添加 App-V 服务器名称条目（“管理模板”>“Windows 组件”>“Windows 远程管理(WinRM)”>“WinRM 客户端”）。

4. Studio 管理员也是 App-V 服务器管理员。
5. 在 App-V 服务器上启用文件共享：在 Windows 资源管理器中或在“运行”命令中输入 `\\<App-V 服务器 FQDN>`。
6. App-V 服务器与 App-V 管理员具有相同的文件共享权限：在 App-V 服务器上，在“存储的用户名和密码”中添加 `\\<App-V 服务器 FQDN>` 条目，并指定在 App-V 服务器上具有管理员权限的用户的凭据。有关指南，请参阅 <http://support.microsoft.com/kb/306541>。
- 如果应用程序发现失败，请检查以下各项：
 1. Studio 管理员是 App-V 管理服务器管理员。
 2. App-V 管理服务器正在运行。请打开 IIS 管理器检查此项。服务器应处于“已启动”和“正在运行”状态。
 3. 在 App-V 服务器上启用 PowerShell 远程处理。如果均未启用，请按照 <http://technet.microsoft.com/en-us/magazine/ff700227.aspx> 中的过程操作。
 4. 软件包具有相应的安全权限，以便 Studio 管理员进行访问。
- 如果 App-V 应用程序未启动，请检查以下各项：
 1. 发布服务器正在运行。请打开 IIS 管理器检查此项。服务器应处于“已启动”和“正在运行”状态。
 2. App-V 软件包具有相应的安全权限，以便用户可进行访问。
 3. 在 VDA 上：
 - 确保 Temp 指向正确的位置，并且 Temp 目录中具有足够的可用空间。
 - 确保已安装 App-V 客户端，并且版本不低于 5.0。
 - 确保您具有管理员权限并运行 `Get-AppvClientConfiguration`。确保将 `EnablePackageScripts` 设置为 1。如果未设置为 1，请运行 `Set-AppvClientConfiguration -EnablePackageScripts $true`。Citrix 建议您在创建主映像时执行此步骤，以便基于主映像创建的所有 VDA 具有正确的配置。
 - 从注册表编辑器 (regedit) 转至 `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Citrix\AppV`。确保 `AppVServers` 注册表项的值采用以下格式：`AppVManagementServer+metadata;PublishingServer`（例如：`http://xmas-demo-appv.blrstrm.com+0+0+1+1+1+0+1;http://xmas-demo-appv.blrstrm.com:8082`）。
 - 确保 `CtxAppVCOMAdmin` 具有管理员权限。`CtxAppVCOMAdmin` 通常在 VDA 安装过程中创建并添加到 VDA 计算机上的本地管理员组。但是，根据 Active Directory 策略，此用户可能会丢失管理关联。
运行 `compmgmt.msc` 并浏览到本地用户和组用户。如果 `CtxAppVCOMAdmin` 不是管理员，请编辑组策略或与管理员联系，以使此用户帐户保留其管理关联。
 4. 在安装 App-V Client 的主映像上，应将 PowerShell ExecutionPolicy 设置为 `RemoteSigned`，因为 Microsoft 提供的 App-V Client 模块未签名，并且此 ExecutionPolicy 允许 PowerShell 运行未签名的本地脚本和 cmdlet。使用以下方法之一设置 ExecutionPolicy：
 - 以管理员身份登录并输入以下 PowerShell cmdlet：`Set-ExecutionPolicy RemoteSigned`。
 - 在“组策略”设置中，转至“计算机配置”>“策略”>“管理模板”>“Windows 组件”>“Windows PowerShell”>“打开脚本执行”。
 5. 检查发布服务器：
 - 运行 `Get-AppvPublishingServer *` 以显示发布服务器的列表。
 - 检查 `UserRefreshonLogon` 是否设置为 `False`。如果不是，则第一个 App-V 应用程序通常无法启动。
 - 在具有管理员权限的情况下，运行 `Set-AppvPublishingServer` 并将 `UserRefreshonLogon` 设置为 `False`。如果这些步骤无法解决问题，则应启用并检查日志。

启用日志

要启用 Studio 日志，请执行以下操作：

1. 创建文件夹 `C:\CtxAppVLogs`。
2. 转至 `C:\ProgramFiles\Citrix\StudioAppVIntegration\SnapIn\Citrix.Appv.Admin.V1`，然后以管理员身份在文本编辑器（例如记事本）中打开 `CtxAppVCommon.dll.config`。注释掉以下行：

要启用 VDA 日志，请执行以下操作：

1. 创建文件夹 `C:\CtxAppVLogs`。
2. 转至 `C:\ProgramFiles\Citrix\Virtual Desktop Agent`，然后以管理员身份在文本编辑器（例如记事本）中打开 `CtxAppVCommon.dll.config`。注释掉以下行：
3. 注释掉以下行并将值字段设置为 1，如下例中所示：

与配置相关的所有日志均位于 `C:\CtxAppVLogs` 中。应用程序启动日志的位置：

- XenDesktop 7.1 和更高版本以及 XenApp 7.5 和更高版本 — `%LOCALAPPDATA%\Citrix\CtxAppVLogs`。
- XenDesktop 7.0 — `%LocalAppData%\temp\CtxAppVLogs`

LOCALAPPDATA 解析为已登录用户的本地文件夹。确保签入启动用户（应用程序启动失败的用户）的本地文件夹。

4. 以管理员身份重新启动 Broker Service 或重新启动 VDA 计算机以启动日志记录。

本地应用程序访问和 URL 重定向

May 04, 2016

本地应用程序访问可将本地安装的 Windows 应用程序无缝集成到托管的桌面环境中，而无需从一台计算机更改为另一台计算机。利用本地应用程序访问，您可以：

- 直接从虚拟桌面访问在物理便携式计算机、PC 或其他设备上本地安装的应用程序。
- 提供灵活的应用程序交付解决方案。如果用户具有您无法虚拟化或 IT 不予维护的本地应用程序，这些应用程序的行为就像安装在虚拟桌面上时一样。
- 通过将已发布应用程序的快捷方式放置在用户的 Windows 设备，当应用程序独立于虚拟桌面托管时可避免双跳延迟。
- 使用如下应用程序：
 - 视频会议软件，例如 GoToMeeting。
 - 尚未虚拟化的特殊或利基应用程序。
 - 采用其他方式时会将大量数据从用户设备传输到服务器再返回用户设备的应用程序和外围设备，例如 DVD 刻录机或 TV 调谐器。

在 XenApp 和 XenDesktop 中，托管桌面会话使用 URL 重定向启动本地应用程序访问应用程序。借助 URL 重定向，可通过多个 URL 地址获得应用程序。通过选择桌面会话中浏览器内部的嵌入式链接，可以启动本地浏览器（根据浏览器的 URL 黑名单）。如果导航至未列入黑名单的 URL，则此 URL 会再次在桌面会话中打开。

URL 重定向仅适用于桌面会话，不适用于应用程序会话。唯一可用于应用程序会话的重定向功能是主机到客户端的内容重定向，它是服务器 FTA 的一种类型。此 FTA 可将某些协议（如 http、https、rtsp 或 mms）重定向到客户端。例如，如果仅使用 http 打开嵌入式链接，这些链接将直接在客户端应用程序中打开。不支持 URL 黑名单或白名单。

启用本地应用程序访问时，对于作为本地运行应用程序、用户托管应用程序中的链接或作为桌面上的快捷方式显示给用户的 URL，将通过以下方式之一进行重定向：

- 从用户的计算机重定向到托管的桌面
- 从 XenApp 或 XenDesktop 服务器重定向到用户计算机
- 在启动（而非重定向）它们的环境中呈现

要指定特定 Web 站点中内容重定向的路径，请在 Virtual Delivery Agent 上配置 URL 白名单和 URL 黑名单。这些名单包含多字符注册表项，用于指定 URL 重定向策略设置；有关详细信息，请参阅“本地应用程序访问策略设置”。

URL 可在 VDA 上呈现，但存在以下例外情况：

- 地理/区域设置信息 — 需要区域设置信息的 Web 站点，如 msn.com 或 news.google.com（根据地理信息打开特定于某个国家/地区的页面）。例如，如果从位于英国的数据中心置备 VDA，而客户端从印度进行连接，则用户将看到 in.msn.com，而不是 uk.msn.com。
- 多媒体内容 — 在客户端设备上呈现包含富媒体内容的 Web 站点时，最终用户将获得本地体验，甚至还可以节省高延迟网络中的带宽。虽然存在 Flash 重定向功能，作为一种补充，带有其他媒体类型（例如，Silverlight）的站点也可以实现重定向。这是一个非常安全的环境。也就是说，管理员批准的 URL 在客户端上运行，而其余 URL 将重定向到 VDA。

除 URL 重定向外，还可以使用文件类型关联 (FTA) 重定向。FTA 在会话中遇到文件时会启动本地应用程序。如果启动本地应用程序，则该应用程序必须具有此文件的访问权限才能将其打开。因此，只能使用本地应用程序打开位于网络共享或客户端驱动器（使用客户端驱动器映射）上的文件。例如，当打开 PDF 文件时，如果 PDF 阅读器是本地应用程序，则文件使用 PDF 阅读器打开。由于本地应用程序可以直接访问文件，因此，无需通过 ICA 网络传输文件，即可打开此文件。

要求、注意事项和限制

本地应用程序访问在面向 VDA for Windows Server OS 和 VDA for Windows Desktop OS 的有效操作系统中受支持，需要

Citrix Receiver for Windows 4.1（最低版本）。支持以下浏览器：

- Internet Explorer 8、9、10 和 11
- Firefox 3.5 至 21.0
- Chrome 10

使用本地应用程序访问和 URL 重定向时请阅读以下注意事项和限制。

- 本地应用程序访问专用于覆盖所有显示器的全屏虚拟桌面，如下所示：
 - 如果本地应用程序访问与在窗口模式下运行或未覆盖所有显示器的虚拟桌面结合使用，用户体验可能会非常混乱。
 - 对于多个显示器，如果最大化其中一个显示器，则该显示器将成为在该会话中启动的所有应用程序的默认桌面，即使随后的应用程序通常在其他显示器中启动也是如此。
 - 此功能支持一个 VDA；不存在与多个并发 VDA 的集成。
- 有些应用程序可能会出现异常行为，对用户产生以下影响：
 - 用户可能对驱动器盘符感到困惑，例如是本地 C:，而不是虚拟桌面 C: 驱动器。
 - 在虚拟桌面中可用的打印机对本地应用程序不可用。
 - 需要提升权限的应用程序不能作为客户端托管应用程序启动。
 - 不会对单实例应用程序（例如 Windows Media Player）进行特殊处理。
 - 本地应用程序随本地计算机的 Windows 主题出现。
 - 不支持全屏应用程序。包括可打开至全屏的应用程序，例如 PowerPoint 幻灯片演示或覆盖整个桌面的照片查看器。
 - 本地应用程序访问复制 VDA 上本地应用程序的属性，例如客户端桌面上的快捷方式和“开始”菜单；但是，不会复制其他属性，例如快捷键和只读属性。
 - 自定义如何处理重叠窗口顺序的应用程序可能会存在不可预测的结果。例如，有些窗口可能会隐藏。
 - 不支持快捷方式，包括我的电脑、回收站、控制面板、网络驱动器快捷方式以及文件夹快捷方式。
 - 不支持以下文件类型和文件：自定义文件类型、没有关联程序的文件、zip 文件和隐藏文件。
 - 不支持对混合的 32 位和 64 位客户端托管应用程序或 VDA 应用程序进行任务栏分组，例如将 64 位 VDA 应用程序和 32 位本地应用程序组合在一起。
 - 不能使用 COM 启动应用程序。例如，如果从 Office 应用程序中单击嵌入式 Office 文档，则检测不到进程启动，且本地应用程序集成失败。
- URL 重定向仅支持显式 URL（即，出现在浏览器的地址栏中或使用浏览器内的导航找到的 URL，具体取决于浏览器）。
- URL 重定向仅适用于桌面会话，不适用于应用程序会话。
- VDA 会话中的本地桌面文件夹不允许用户创建新文件。
- 对于本地运行的应用程序的多个实例而言，其行为方式取决于为虚拟桌面建立的任务栏设置。但是，本地运行应用程序的快捷方式不与这些应用程序的运行实例一起分组，也不与托管应用程序的运行实例或托管应用程序的固定快捷方式一起分组。用户只能从任务栏关闭本地运行的应用程序的窗口。尽管用户可以将本地应用程序窗口固定在桌面任务栏和“开始”菜单中，但使用这些快捷方式时，不一定总是可以启动这些应用程序。

与 Windows 交互

本地应用程序访问与 Windows 的交互包括以下行为：

- Windows 8 和 Windows Server 2012 快捷方式行为
 - 客户端上安装的 Windows 应用商店应用程序并不随本地应用程序访问快捷方式进行枚举。
 - 默认情况下，通常使用 Windows 应用商店应用程序打开图像和视频文件。但是，本地应用程序访问会枚举 Windows 应用商店应用程序，并使用桌面应用程序打开快捷方式。
- 本地程序
 - 对于 Windows 7，可从开始菜单中访问此文件夹。
 - 对于 Windows 8，仅当用户从“开始”屏幕中选择所有应用程序类别时，本地程序才可用。并非所有子文件夹均显示在本地程序中。
- 针对应用程序的 Windows 8 图形功能

- 桌面应用程序限制在桌面区域内，并被“开始”屏幕和 Windows 8 风格应用程序所覆盖。
- 在多显示器模式下，本地应用程序访问应用程序与桌面应用程序的行为有所不同。在多显示器模式下，“开始”屏幕和桌面显示在不同的显示器中。
- Windows 8 和本地应用程序访问 URL 重定向
 - 由于 Windows 8 Internet Explorer 未启用任何加载项，因此使用桌面 Internet Explorer 启用 URL 重定向。
 - 在 Windows Server 2012 中，Internet Explorer 默认情况下禁用加载项。要实现 URL 重定向，禁用 Internet Explorer 增强的配置。重置 Internet Explorer 选项并重新启动，以确保为标准用户启用加载项。

配置本地应用程序访问和 URL 重定向

May 04, 2016

要将本地应用程序访问和 URL 重定向用于 Citrix Receiver，请执行以下操作：

- 在本地客户端计算机上安装 Receiver。可以在 Receiver 安装期间启用这两项功能，或者使用组策略编辑器启用本地应用程序访问模板。
- 将允许本地应用程序访问策略设置为启用。也可以为 URL 重定向配置 URL 白名单和黑名单策略。有关详细信息，请参阅[本地应用程序访问策略设置](#)。

为所有本地应用程序启用本地应用程序访问和 URL 重定向：

1. 将允许本地应用程序访问策略设置为启用。启用此设置时，VDA 让客户端决定在会话中是否启用管理员发布的应用程序快捷方式和本地应用程序访问快捷方式。（禁用此设置时，管理员发布的应用程序快捷方式和本地应用程序访问快捷方式不适用于 VDA。）此策略设置适用于整个计算机，URL 重定向策略也是如此。
2. 在安装 Citrix Receiver 期间为计算机上的所有用户启用本地应用程序访问和 URL 重定向。此操作还会注册 URL 重定向所需的浏览器加载项。

从命令提示窗口，使用以下选项运行相应的命令以安装 Receiver：

```
CitrixReceiver.exe /ALLOW_CLIENTHOSTEDAPPSURL=1
```

```
CitrixReceiverWeb.exe /ALLOW_CLIENTHOSTEDAPPSURL=1
```

1. 运行 gpedit.msc。
2. 选择计算机配置。在管理模板上单击鼠标右键，然后依次选择添加/删除模板 > 添加。
3. 添加位于 Receiver Configuration 文件夹中的 icaclient.adm 模板（通常位于 c:\Program Files (x86)\Citrix\Online Plugin\Configuration）。（将 icaclient.adm 模板添加到计算机配置中以后，在用户配置中也可以使用此模板。）
4. 依次展开管理模板 > 经典管理模板(ADM) > Citrix 组件 > Citrix Receiver > 用户体验。
5. 选择本地应用程序访问设置。
6. 选择启用，然后选择允许 URL 重定向。对于 URL 重定向，按照如下所述使用命令行注册浏览器加载项。

仅提供对已发布应用程序的访问：

1. 在安装 Delivery Controller 的服务器上，运行 regedit.exe。
 1. 导航到 HKLM\Software\Wow6432Node\Citrix\DesktopStudio。
 2. 添加 REG_DWORD 条目 ClientHostedAppsEnabled，值为 1。（值为 0 表示禁用本地应用程序访问。）
2. 依次重新启动 Delivery Controller 服务器和 Studio。
3. 发布本地应用程序访问应用程序。
 1. 在 Studio 导航窗格中选择交付组，然后选择“应用程序”选项卡。
 2. 在“操作”窗格中选择创建本地访问应用程序。
 3. 选择桌面交付组。
 4. 在用户的本地计算机上输入应用程序的完整可执行路径。
 5. 指出虚拟桌面上的本地应用程序快捷方式是否显示在“开始”菜单、桌面或二者上。
 6. 接受名称页面上的默认值，然后检查设置。
4. 在安装 Citrix Receiver 期间为计算机上的所有用户启用本地应用程序访问和 URL 重定向。此操作还会注册 URL 重定向所需的浏览器加载项。

从命令提示窗口，使用以下选项运行命令以安装 Receiver：

CitrixReceiver.exe /ALLOW_CLIENTHOSTEDAPPSURL=1

CitrixReceiverWeb.exe /ALLOW_CLIENTHOSTEDAPPSURL=1

5. 将允许本地应用程序访问策略设置为“启用”。启用此设置时，VDA 让客户端决定在会话中是否启用管理员发布的应用程序快捷方式和本地应用程序访问快捷方式。（禁用此设置时，管理员发布的应用程序快捷方式和本地应用程序访问快捷方式不适用于 VDA。）

注意：使用 /ALLOW_CLIENTHOSTEDAPPSURL=1 选项从命令行安装 Receiver 时，会自动注册 URL 重定向所需的浏览器加载项。

可以使用以下命令注册和取消注册一个或所有加载项：

- 在客户端设备上注册加载项：<客户端安装文件夹>\redirector.exe /reg<浏览器>
- 在客户端设备上取消注册加载项：<客户端安装文件夹>\redirector.exe /unreg<浏览器>
- 在 VDA 上注册加载项：<VDA 安装文件夹>\VDARedirector.exe /reg<浏览器>
- 在 VDA 上取消注册加载项：<VDA 安装文件夹>\VDARedirector.exe /unreg<浏览器>

其中，<浏览器> 可以是 IE、FF、Chrome 或 All。

例如，以下命令在运行 Receiver 的设备上注册 Internet Explorer 加载项。

C:\Program Files\Citrix\ICA Client\redirector.exe/regIE

以下命令在 Windows Server OS VDA 上注册所有加载项。

C:\Program Files (x86)\Citrix\System32\VDARedirector.exe /regAll

说明	配置
默认情况下，Internet Explorer 重定向所输入的 URL。如果 URL 未列入黑名单中，但是通过浏览器或 Web 站点重定向到其他 URL，则即使最终 URL 在黑名单中，也不会重定向此 URL。	为使 URL 重定向正常运行，请在浏览器提示时启用加载项。如果禁用使用 Internet 选项的加载项或提示中的加载项，URL 重定向将无法正常运行。
Firefox 加载项始终重定向 URL。	安装加载项时，Firefox 在新的选项卡页面上提示是否允许/阻止安装加载项。必须允许加载项，才能正常使用此功能。
Chrome 加载项始终重定向导航到的最终 URL，而非输入的 URL。	已在外部安装扩展。如果禁用扩展，URL 重定向功能在 Chrome 中将无法正常使用。如果在隐身模式中需要 URL 重定向，请在浏览器的“设置”中允许扩展在该模式下运行。

1. 在托管桌面上，运行 gpedit.msc。

1. 导航至 HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\Client Hosted Apps\Policies\Session State。
对于 64 位系统，导航至 HKEY_LOCAL_MACHINE\SOFTWARE\wow6432node\Citrix\Client Hosted Apps\Policies\Session State。

2. 添加 REG_DWORD 条目 Terminate，并采用以下其中一个值：

- 1 - 当用户从虚拟桌面注销或断开连接时，本地应用程序继续运行。一旦重新建立连接，如果本地应用程序在本地环境中可用，将重新集成。
- 3 - 当用户从虚拟桌面注销或断开连接时，本地应用程序关闭。

服务器 VDI

Nov 02, 2016

通过“服务器 VDI”（虚拟桌面基础结构）功能，可以从服务器操作系统为单个用户交付桌面。

- 企业管理员可以将服务器操作系统作为 VDI 桌面进行交付，这对于工程师和设计师等用户非常有帮助。
- 服务提供商可以利用云提供桌面；这些桌面受 Microsoft 服务提供商许可协议 (SPLA) 约束。

通过 Citrix 策略设置“增强的桌面体验”，可以使服务器操作系统像 Windows 7 操作系统一样运行。

服务器 VDI 不支持以下功能：

- Personal vDisk
- HDX 3D Pro
- 托管应用程序
- 本地应用程序访问
- 直接（非代理）桌面连接
- Remote PC Access

要使服务器 VDI 能够与扫描仪等 TWAIN 设备结合使用，必须安装 Windows Server 桌面体验功能。在 Windows Server 2012 中，这是一项可以从“管理工具”>“服务器管理器”>“功能”>“添加功能”>“桌面体验”中安装的可选功能。

服务器 VDI 与 VDA for Windows Server OS 受相同的服务器操作系统支持。

1. 安装前的 Windows 服务器准备工作：确保未安装远程桌面服务角色服务，并且将用户限定于单个会话：
 - 通过 Windows 服务器管理器，确保未安装远程桌面服务角色服务。如果先前已安装这些服务，请将其删除。
 - 确保已启用“限制每个用户只能进行一个会话”属性。
 - 在 Windows Server 2008 R2 上，可通过访问“管理工具”>“远程桌面服务”>“远程桌面会话主机配置”。在“编辑”设置 >“常规”部分，Restrict each user to a single session（限制每个用户只能进行一个会话）设置应为是。
 - 在 Windows Server 2012 上，编辑注册表，并设置终端服务器设置。在注册表项 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TerminalServer 中，将 DWORD fSingleSessionPerUser 设置为 1。
2. 对于 Windows Server 2008 R2，在安装 VDA 之前，在服务器上安装 Microsoft .NET Framework 3.5 SP1。
3. 使用命令行接口在支持的服务器或服务器主映像上安装 VDA，同时指定 /quiet 和 /servervdi 选项。（默认情况下，安装程序会阻止在服务器操作系统上安装 Windows 桌面操作系统 VDA；可通过命令行覆盖此行为。）
XenDesktopVdaSetup.exe /quiet /servervdi

使用命令行安装 VDA 时，可以通过 **/controllers** 选项指定一个或多个 Delivery Controller。

使用 **/enable_hdx_ports** 选项打开防火墙中的端口，除非要手动配置防火墙。

如果要在映像上安装 VDA，并使用 MCS 从该映像创建服务器 VM，请添加 **/masterimage** 选项。

请不要包含服务器 VDI 不支持的功能所对应的选项，例如 /baseimage、/enable_hdx_3d_pro 或 /xa_server_location。

4. 为服务器 VDI 创建计算机目录。
 1. 在操作系统页面上，选择 Windows 桌面操作系统。
 2. 在摘要页面上，为管理员指定可明确将其标识为服务器 VDI 的计算机目录名称和说明；在 Studio 中，只能通过该内容来指示目录支持服务器 VDI。在 Studio 中使用搜索功能时，所创建的服务器 VDI 目录将显示在“桌面操作系统计算机”选项卡上，即使此 VDA 安装在服务器上也是如此。
5. 创建交付组，并为其分配在上一步中所创建的服务器 VDI 目录。

如果安装 VDA 时未指定 Delivery Controller，之后可使用 Citrix 策略设置、Active Directory 或通过编辑 VDA 计算机的注册表值进行指定。

删除组件

May 04, 2016

要删除组件，Citrix 建议使用专门用于删除或更改程序的 Windows 功能。也可以使用命令行或安装介质中的脚本删除组件。

删除组件时，不会删除必备项，也不会更改防火墙设置。删除 Controller 时，不会删除 SQL Server 软件和数据库。

删除 Controller 之前，请先将其从站点中删除。删除 Studio 或 Director 之前，Citrix 建议先将其关闭。

如果从包含 Web Interface 的早期部署升级了 Controller，必须单独删除 Web Interface 组件；不能使用安装程序删除 Web Interface。

使用专门用于删除或更改程序的 Windows 功能：

- 要删除 Controller、Studio、Director、许可证服务器或 StoreFront，请选择 Citrix XenApp <版本> 或 Citrix XenDesktop <版本>，然后单击鼠标右键并选择卸载。此时将启动安装程序，从中可选择要删除的组件。
也可以通过在 Citrix StoreFront 上单击鼠标右键并选择卸载来删除 StoreFront。
- 要删除 VDA，请选择 Citrix Virtual Delivery Agent <版本>，然后单击鼠标右键并选择卸载。此时将启动安装程序，从中可选择要删除的组件。
- 要删除通用打印服务器，请选择 Citrix 通用打印服务器，然后单击鼠标右键并选择卸载。

从安装介质上的 \x64\XenDesktop Setup 目录中，运行 XenDesktopServerSetup.exe 命令。

- 要删除一个或多个组件，请使用 /remove 和 /components 选项。
- 要删除所有组件，请使用 /removeall 选项。

有关命令和参数的详细信息，请参阅[使用命令行安装](#)。

例如，以下命令可删除 Studio。

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

从安装介质上的 \x64\XenDesktop Setup 目录中，运行 XenDesktopVdaSetup.exe 命令。

- 要删除一个或多个组件，请使用 /remove 和 /components 选项。
- 要删除所有组件，请使用 /removeall 选项。

有关命令和参数的详细信息，请参阅[使用命令行安装](#)。

例如，以下命令可删除 VDA 和 Receiver。

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

要使用 Active Directory 中的脚本删除 VDA，请参阅[使用脚本安装或删除 Virtual Delivery Agent](#)。

升级和迁移

May 04, 2016

通过升级可以将部署更改为最新的组件版本，而无需设置新的计算机或站点，这称为原位升级。可以执行以下升级：

- 从 XenDesktop 5（或更高版本）升级到 XenDesktop 7.6
- 从 XenApp 7.5 升级到 XenApp 7.6

也可以将 XenApp 6.5 工作服务器升级到 XenApp 7.6 VDA for Windows Server OS。这是迁移到 XenApp 6.5 的补充活动。

要升级 XenDesktop 5（或更高版本）场或 XenApp 7.5 站点，请执行以下操作：

1. 在安装核心组件和 VDA 的计算机上运行安装程序。软件会确定是否有可用的升级并安装更新的版本。
2. 使用新升级的 Studio 来升级数据库和站点。

有关详细信息，请参阅[升级部署](#)。

有关安装 Controller 修补程序的详细信息，请参阅知识中心文章 [CTX201988](#)。

要将 XenApp 6.5 工作服务器升级到 XenApp 7.6 VDA，请执行以下操作：

1. 在 XenApp 6.5 工作服务器上运行产品安装程序。此软件从 XenApp 6.5 场删除服务器，删除 XenApp 6.5 软件，并安装 7.6 版 VDA for Windows Server OS。
2. 升级服务器后，将其添加到 7.6 站点中的计算机目录和交付组。

有关详细信息，请参阅[将 XenApp 6.5 工作进程升级至新的 VDA for Windows Server OS](#)。

将数据从早期部署迁移到最新版本。可以迁移 XenApp 6.5 或 XenDesk 4 部署。迁移包括安装 7.6 组件和创建新站点，从旧场导出数据，然后将数据导入到新站点。

要从 XenApp 6.5 进行迁移，请执行以下操作：

1. 安装核心组件并创建新的 XenApp 站点。
2. 从 XenApp 6.5 控制器，使用 PowerShell cmdlet 将策略和/或场数据导出到 XML 文件中。您可以编辑 XML 文件的内容以定制要导入的信息。
3. 从新的 7.6 站点，使用 PowerShell cmdlet 和 XML 文件将策略和/或应用程序数据导入到新站点。
4. 在新站点上完成迁移后需执行的任务。

有关详细信息，请参阅[迁移 XenApp 6.x](#)。

要从 XenDesktop 4 进行迁移，请执行以下操作：

1. 安装核心组件并创建新的 XenDesktop 站点。
2. 从 XenDesktop 4 场，使用导出命令工具将场数据导出到 XML 文件。您可以编辑 XML 文件的内容以定制要导入的信息。
3. 从 7.6 站点，使用导入命令工具和 XML 文件将场数据导入到新站点。
4. 在新站点上完成迁移后需执行的任务。

有关详细信息，请参阅[迁移 XenDesktop 4](#)。

升级部署

Nov 02, 2016

You can upgrade certain deployments to newer versions without having to first set up new machines or Sites; this is called an in-place upgrade. You can upgrade:

- From XenDesktop version 5 (or a later version) to the latest released (current) XenDesktop version
- From XenApp version 7.5 (or a later version) to the latest released (current) XenApp version

You can also use the XenApp 7.6 installer to upgrade a XenApp 6.5 worker server to a XenApp 7.6 VDA for Windows Server OS. This is a supplementary activity to migrating XenApp 6.5; see [Upgrade a XenApp 6.5 worker to a new VDA for Windows Server OS](#).

To start an upgrade, you run the installer from the new version to upgrade previously installed core components (Delivery Controller, Citrix Studio, Citrix Director, Citrix License Server) and VDAs. The installer determines which components require upgrading and then starts the upgrade at your command. After upgrading the components, you use the newly upgraded Studio to upgrade the Site database and the Site.

In this content, the word product refers to XenApp 7.x or XenDesktop 7.x, unless otherwise noted.

You cannot upgrade:

- From an Early Release or Technology Preview version
- From a XenApp version earlier than 7.5 (except replacing XenApp 6.5 software on a server with a current VDA for Server OS; see [Migrate XenApp 6.x](#))
- From a XenDesktop version earlier than 5.6; see [Migrate XenDesktop 4](#)
- From XenApp to XenDesktop, or from XenDesktop to XenApp

Using the product installer and Studio, you can upgrade:

- Delivery Controllers 5 or later
- VDA 5.0 SP1 or later
 - Unlike earlier VDA releases, you must use the product installer to upgrade VDAs; you cannot use MSIs.
 - If the installer detects Receiver for Windows (Receiver.exe) on the machine, it is upgraded to the Receiver version included on the product installation media.
 - If the installer detects Receiver for Windows Enterprise (CitrixReceiverEnterprise.exe) on the machine, it is upgraded to Receiver for Windows Enterprise 3.4.
- Director 1 or later
- Database - This upgrades the schema and migrates data for the Site database (plus the Configuration Logging and Monitoring databases, if you're upgrading from an earlier 7.x version)
- Personal vDisk

Using the guidance in the feature/product documentation, upgrade the following if needed:

- Provisioning Services (for XenApp 7.x and XenDesktop 7.x, Citrix recommends using the latest released version; the minimum supported version is Provisioning Services 7.0).
 - Upgrade the Provisioning Services server using the server rolling upgrade, and the clients using vDisk versioning.
 - Provisioning Services 7.x does not support creating new desktops with XenDesktop 5 versions. So, although existing desktops will continue to work, you cannot use Provisioning Services 7.x to create new desktops until you upgrade XenDesktop. Therefore, if you plan a mixed environment of XenDesktop 5.6 and 7.x Sites, do not upgrade Provisioning

Services to version 7.

- Microsoft System Center Virtual Machine Manager SCVMM. The current product supports SCVMM 2012 and SCVMM 2012 SP1; XenDesktop 5.x supports earlier versions. Use the following upgrade sequence to avoid downtime:
 1. If you have Controllers running versions earlier than XenDesktop 5.6 FP1, upgrade them to XenDesktop 5.6 FP1 (see the XenDesktop documentation for that version).
 2. Upgrade the SCVMM server to SCVMM 2012; see the Microsoft documentation for instructions.
 3. Upgrade XenDesktop components to the current version.
 4. Optionally, upgrade the SCVMM server to SCVMM 2012 SP1.
- StoreFront.

•	
	You must use the product installer's graphical or command-line interface to upgrade core components and VDAs; you cannot import or migrate data from an earlier version.
	If you install or upgrade any components to the new version but choose not to upgrade other components (on different machines) that require upgrade, Studio will remind you. For example, if an upgrade includes new versions of the Controller and Studio, and you upgrade only the Controller (but you do not run the installer on the machine where Studio is installed), Studio will not let you continue to manage the Site until you upgrade Studio.
	Before upgrading the Citrix License Server, be sure your Subscription Advantage date is valid for the new product version. If you are upgrading from an earlier 7.x product version, the date must be at least 2013.0522.
	You cannot upgrade XenDesktop Express Edition. Obtain and install a license for a currently supported edition, then upgrade it.
	Before beginning any upgrade activity, back up the database, as described in CTX135207 , so you can restore it if any issues are discovered after the database upgrade.
	Optionally, back up templates and upgrade hypervisors, if used.
	If you must continue to run earlier version Sites and current version Sites, see Mixed environment considerations .
	If you have VDAs installed on Windows XP or Windows Vista machines, see VDAs on machines running Windows XP or Windows Vista .
	If you do not plan to upgrade all VDAs to the latest version, review Mixed VDA support .

•	If your deployment includes Web Interface, Citrix recommends using StoreFront.
	In addition to being a domain user, you must be a local administrator on the machines where you are upgrading product components.
	When you upgrade, you do not choose or specify the product (XenApp or XenDesktop), which was set during the initial installation.
	Review the upgrade sequence below so you can plan for and mitigate potential outages.

When your environment contains Sites/farms with different product versions (a mixed environment), Citrix recommends using StoreFront to aggregate applications and desktops from different product versions (for example, if you have a XenDesktop 7.1 Site and a XenDesktop 7.5 Site). For details, see the StoreFront documentation.

- Generally, the current Studio and Director versions manage/monitor only current Sites. (Although this version of Director can monitor XenDesktop 5.x VDAs, some data (including logon duration) will not be available for those VDAs.) For example, you cannot manage a XenDesktop 7.1 Site with Studio version 7.6. Similarly, you cannot manage a XenDesktop 7.6 Site with a Studio version 7.1.
- You can use current VDAs in deployments containing earlier Controller versions. Keep in mind that in such cases, new features in the current release may not be available. See
— *Mixed VDA considerations*
below.
- Sites with Controllers at version 5.x and VDAs at version 7.x should remain in that state only temporarily. Ideally, you should complete the upgrade of all components as soon as possible.
- In a mixed environment, continue using the Studio and Director versions for each release, but make sure that different versions are installed on separate machines.
- If you plan to run XenDesktop 5.6 and 7.x Sites simultaneously and use Provisioning Services for both, either deploy a new Provisioning Services for use with the 7.x Site, or upgrade the current Provisioning Services and be unable to provision new workloads in the XenDesktop 5.6 Site.
- Do not upgrade a standalone Studio version until you are ready to use the new version.

You cannot upgrade VDAs installed on machines running Windows XP or Windows Vista to a 7.x version. You must use VDA 5.6 FP1 with certain hotfixes; see [CTX140941](#) for instructions. Although earlier-version VDAs will run in a 7.x Site, they cannot use many of its features, including:

- Features noted in Studio that require a newer VDA version.
- Configuring App-V applications from Studio.
- Configuring Receiver StoreFront addresses from Studio.
- Automatic support for Microsoft Windows KMS licensing when using Machine Creation Services. See [CTX128580](#).
- Information in Director:
 - Logon times and logon end events impacting the logon duration times in the Dashboard, Trends, and User Detail views.
 - Logon duration breakdown details for HDX connection and authentication time, plus duration details for profile load, GPO load, logon script, and interactive session establishment.
 - Several categories of machine and connection failure rates.

- Activity Manager in the Help Desk and User Details views.

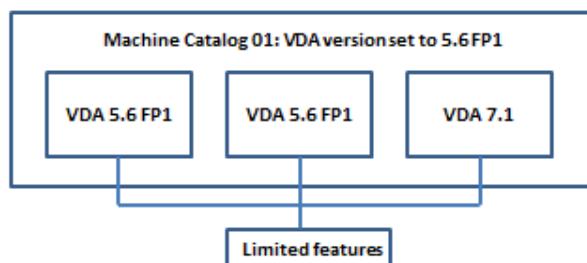
Citrix recommends reimaging Windows XP and Windows Vista machines to a supported operating system version and then installing the latest VDA.

To upgrade VDAs installed on machines running Windows 8.x or Windows 7 to Windows 10, Citrix recommends reimaging Windows 7 and Windows 8.x machines to Windows 10 and then installing the supported VDA for Windows 10, using the standalone VDA installation package delivered with XenApp and XenDesktop 7.6 FP3. If reimaging is not an option, uninstall the VDA prior to upgrading the operating system, otherwise the VDA will be in an unsupported state.

When you upgrade the product to a later version, Citrix recommends you upgrade all the core components and VDAs so you can access all the new and enhanced features in your edition. For example, to use the session prelaunch, session linger, and unauthenticated users features in the 7.6 release, the VDAs must have a minimum version of 7.6 installed.

In some environments, you may not be able to upgrade all VDAs to the most current version. In this scenario, when you create a machine catalog, you can specify the VDA version installed on the machines. By default, this setting specifies the latest recommended VDA version; you need to consider changing this setting only if the machine catalog contains machines with earlier VDA versions. However, mixing VDA versions in a machine catalog can have unintended effects

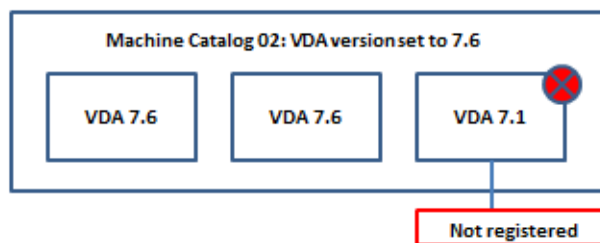
As noted above, if your deployment includes Windows XP and Windows Vista systems, you must use an earlier VDA version, and the machine catalog containing those machines must specify VDA version 5.6 FP1. The VDAs will register successfully with the Controller, but those machines will be unable to use many of the new features in the 7.x versions (including StoreFront). This also applies to any machines you add to that catalog that have 7.x version VDAs. The following graphic illustrates this.



In the above case, if you must continue to use older VDAs, place them in a machine catalog by themselves.

If a machine catalog is created with the default recommended VDA version setting, and any of the machines in the catalog has an earlier VDA version installed, those machines will not be able to register with the Controller and will not work.

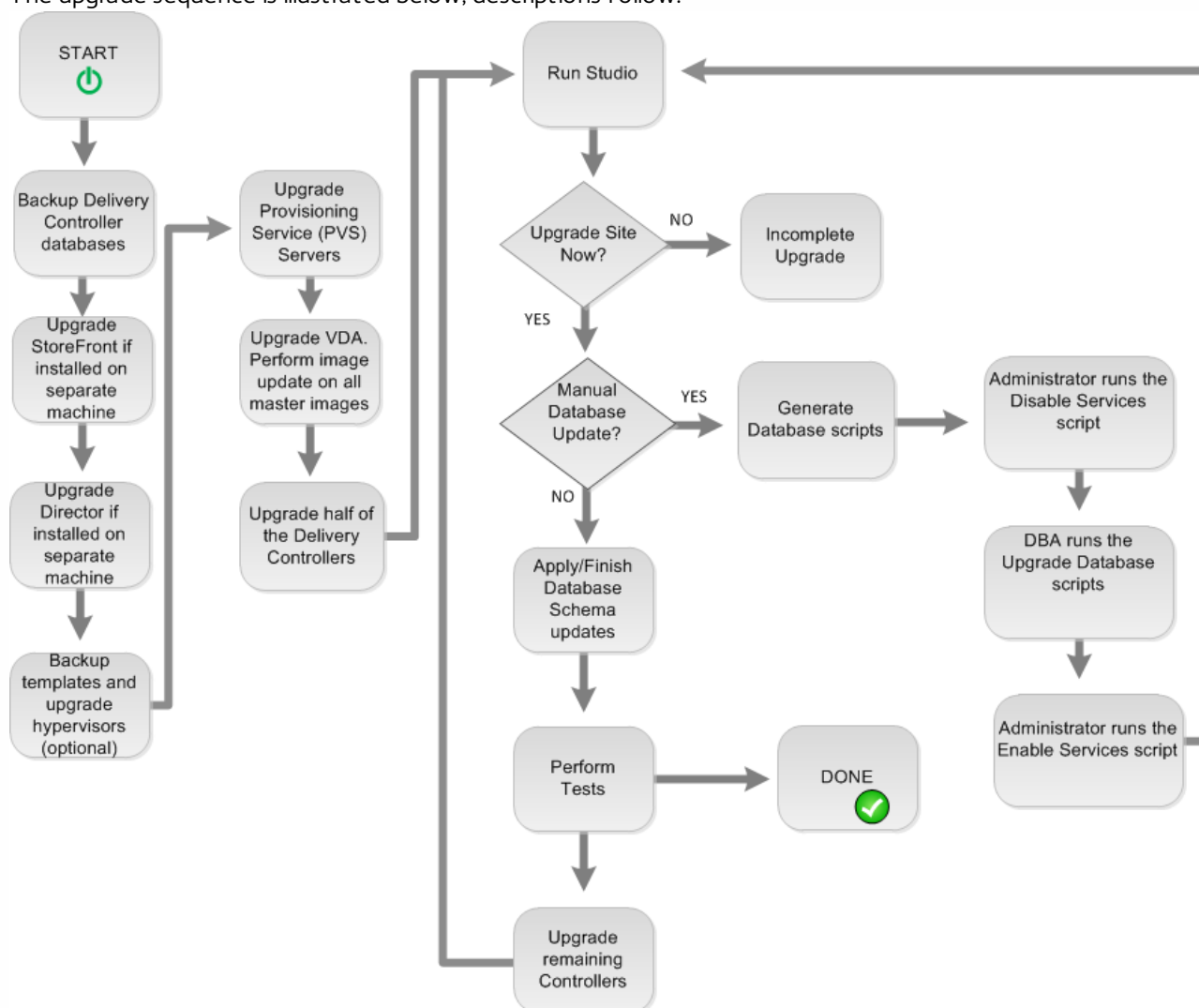
For example, assume the most recent VDA version is 7.6. You create a machine catalog with the default VDA setting: "7.6 (recommended, to access the latest features)." You add three machines to that catalog: two with VDA 7.6 and one with VDA 7.1.



In this example, the machine with VDA 7.1 will not register with the Controller. If you cannot upgrade that VDA, consider creating a separate machine catalog configured with a VDA setting of "version 7.0 or later" and adding that machine. Although that machine will not be able to take advantage of new 7.6 features, it will be able to register with the Controller.

If components are installed on different machines, you run the installer on each of those machines.

The upgrade sequence is illustrated below; descriptions follow.



To run the product installer graphical interface, log on to the machine and then insert the media or mount the ISO drive for the new release. Double-click AutoSelect. To use the command-line interface, see

— *Install using the command line*

1. If more than one core component is installed on the same server (for example, the Controller, Studio, and License Server)

and several of those components have new versions available, they will all be upgraded when you run the installer on that server. If any core components are installed on machines other than the Controller, run the installer on each of those machines (in the preferred order: License Server, StoreFront, and then Director).

2. Upgrade the Provisioning Services servers and clients, using the guidance in the Provisioning Services documentation.
3. Run the product installer on machines containing VDAs. Although you can upgrade VDAs before or after upgrading the Controllers, Citrix recommends you do so before, because it allows you to quickly enable new features after the upgrade. When upgrading VDAs from an earlier 7.x version that are installed on physical machines (including Remote PC Access), use the command-line interface with the parameter: `/EXCLUDE "Personal vDisk","Machine Identity Service"`. For example: `C:\x64\XenDesktop Setup\XenDesktopVdaSetup.exe /EXCLUDE "Personal vDisk","Machine Identity Service"`
4. Run the product installer on half of the Controllers. (This will also upgrade any other core components installed on those servers.) For example, if your Site has four Controllers, run the installer on two of them.
 - Leaving half of the Controllers active allows users to access the Site. VDAs can register with the remaining Controllers. There may be times when the Site has reduced capacity because fewer Controllers are available. The upgrade causes only a brief interruption in establishing new client connections during the final database upgrade steps. The upgraded Controllers cannot process requests until the entire Site is upgraded.
 - If your Site has only one Controller, the Site is inoperable during the upgrade.
5. If Studio is installed on a different machine than one of the Controllers you upgraded in the previous step, run the installer on the machine where Studio is installed.
6. From the newly upgraded Studio, upgrade the Site database. For details, see [Upgrade the database and Site](#).
7. From the newly upgraded Studio, select Citrix Studio site-name in the navigation pane. Select the Common Tasks tab. Select Upgrade remaining Delivery Controllers.
8. After completing the upgrade and confirming completion, close and then reopen Studio.
9. In the Site Configuration section of the Common Tasks page, select Perform registration. Registering the Controllers makes them available to the Site.
10. After you select Finish when the upgrade completes, you are offered the opportunity to enroll in the Citrix Customer Experience Improvement Program (CEIP), which collects anonymous information about your deployment. That information is then used to improve product quality, reliability, and performance.
11. After upgrading components, the database, and the Site, use Studio to:
 - Test the newly-upgraded Site. From Studio, select Citrix Studio site-name in the navigation pane. Select the Common Tasks tab and then select Test Site. These tests were run automatically after you upgraded the database, but you can run them again at any time.
 - Update all master images that use the upgraded VDA.
 - Upgrade machine catalogs and Delivery Groups.

Upgrade the database and Site

After upgrading the core components and VDAs, use the newly upgraded Studio to initiate an automatic or manual database and Site upgrade.

- For an automatic database upgrade, the Studio user's permissions must include the ability to update the SQL Server database schema (for example, the db_securityadmin or db_owner database role).
- If the Studio user does not have those permissions, initiating a manual database upgrade will generate scripts. The Studio user runs some of the scripts from Studio; the database administrator runs other scripts using a tool such as SQL Server Management Studio. If the SQL scripts are run manually, they should be run using either the SQLCMD utility or using the SQL Management Studio in SQLCMD mode. Inaccurate errors may result otherwise.

Important: Citrix strongly recommends you back up the database before upgrading, as described in [CTX135207](#). During a database upgrade, product services are disabled. During that time, Controllers cannot broker new connections for the Site, so plan carefully.

After the database upgrade completes and product services are enabled, Studio tests the environment and configuration, and then generates an HTML report. If problems are identified, you can restore the database backup. After resolving issues, you can upgrade the database again.

Upgrade the database and Site automatically - Launch the newly upgraded Studio. After you choose to start the Site upgrade automatically and confirm that you are ready, the database and Site upgrade proceeds.

Upgrade the database and Site manually - This process includes generating and running scripts.

1. Launch the newly upgraded Studio. After you choose to manually upgrade the Site, the wizard checks for License Server compatibility and requests confirmation. After you confirm that you have backed up the database, the wizard generates and displays the scripts and a checklist of upgrade steps.
2. Run the following scripts in the order shown:

Script	Description
DisableServices.ps1	PowerShell script to be run by the Studio user on a Controller to disable product services.
UpgradeSiteDatabase.sql	SQL script to be run by the database administrator on the server containing the Site database, using a tool such as SQL Server Management Studio.
UpgradeMonitorDatabase.sql	SQL script to be run by the database administrator on the server containing the Monitor database, using a tool such as SQL Server Management Studio.
UpgradeLoggingDatabase.sql	SQL script to be run by the database administrator on the server containing the Configuration Logging database, using a tool such as SQL Server Management Studio. Run this script only if this database changes (for example, after applying a hotfix).
EnableServices.ps1	PowerShell script to be run by the Studio user on a Controller to enable product services.

3. After you complete the displayed checklist tasks, select Finish upgrade.

将 XenApp 6.5 工作进程升级至新的 VDA for Windows Server OS

May 04, 2016

在 XenApp 6.5 工作服务器上运行 XenApp 7.6 安装程序时，此安装程序将：

- 从 XenApp 6.5 场删除服务器（此任务自动调用 XenApp 6.5 安装程序的命令行接口）
- 删除 XenApp 6.5 软件
- 安装新的（XenApp 7.6 或受支持的更高版本）VDA for Windows Server OS

使用安装程序的图形界面时，所遵循的向导与在新 XenApp 站点中安装 VDA for Windows Server OS 时使用的向导相同。同样，命令行接口使用的命令和参数与安装其他 VDA 时使用的命令和参数相同。

您可能已经熟悉使用安装程序安装 XenApp 7.6 核心组件和其他 VDA 的操作。要了解准备信息，请参阅 [VDA 安装准备](#)。然后，启动安装程序（[使用图形界面进行安装](#)）或在 XenApp 6.5 工作服务器上发布命令（[使用命令行安装 VDA](#)）。

须知事项：

- 此升级在仅会话主机模式下配置的 XenApp 6.5 服务器（也称为仅会话或工作服务器）上有效。
- 卸载 XenApp 6.5 需要重新启动多个服务器。使用命令行接口时，可以使用 /NOREBOOT 选项禁止自动操作；但是，必须重新启动服务器才能使卸载和后续安装生效。
- 如果卸载 XenApp 过程中出错，请检查错误消息中引用的卸载错误日志。卸载日志文件位于“%TEMP%\Citrix\XenDesktop Installation\XenApp 6.5 Uninstall Log Files\”文件夹中。
- 升级 XenApp 6.5 工作服务器后，从新 XenApp 站点中的 Studio 中，为升级后的工作进程创建计算机目录（或编辑现有目录）。
- 如果从 XenApp 6.5 控制器服务器迁移策略和应用程序设置（请参阅[迁移 XenApp 6.x](#)），请将包含已迁移的发布应用程序的交付组分配到在 XenApp 6.5 中托管这些应用程序的计算机目录。

症状：删除 XenApp 6.5 软件失败。卸载日志包含消息：“Error 25703.An error occurred while plugging XML into Internet Information Server.Setup cannot copy files to your IIS Scripts directory.Please make sure that your IIS installation is correct.”（错误 25703。将 XML 插入 Internet Information Server 时出错。安装程序无法将文件复制到 IIS 脚本目录。请确保 IIS 安装正确无误。）

- 原因：在以下情况下系统上会出现此问题：(1) 首次安装 XenApp 6.5 时，您指示 Citrix XML Service (CtxHttp.exe) 不能与 IIS 共享一个端口，以及 (2) 安装了 .NET Framework 3.5.1。
- 解决方案：
 1. 使用 Windows 删除服务器角色向导删除 Web 服务器 (IIS) 角色。（之后可以重新安装 Web 服务器 (IIS) 角色。）
 2. 重新启动服务器。
 3. 使用“添加/删除程序”卸载以下程序：
 1. Citrix XenApp 6.5
 2. Microsoft Visual C++ 2005 可再发行组件包 (x64)，版本 8.0.56336
 4. 重新启动服务器。
 5. 运行 XenApp 7.6 安装程序以安装 VDA for Windows Server OS。

迁移 XenApp 6.x

May 04, 2016

重要：请在开始迁移之前完整阅读本文。

XenApp 6.x 迁移工具（简称迁移工具）是关于 PowerShell 脚本的集合，这些脚本中包含用于迁移 XenApp 6.x（6.0 或 6.5）策略和场数据的 cmdlet。在 XenApp 6.x 控制器服务器上，运行导出 cmdlet 以将数据收集到 XML 文件中。然后，从 XenApp 7.6 控制器，运行导入 cmdlet，以使用在导出过程中收集的数据创建对象。

[此处](#)提供概述迁移工具的视频。

以下操作顺序概述了迁移过程，之后将提供详细信息。

- 在 XenApp 6.0 或 6.5 控制器上：
 - 导入 PowerShell 导出模块。
 - 运行导出 cmdlet 以将策略和/或场数据导入到 XML 文件中。
- 将 XML 文件和图标文件夹（如果在导出过程中选择不将图标文件夹嵌入到 XML 文件中）复制到 XenApp 7.6 控制器。
- 在 XenApp 7.6 控制器上：
 - 导入 PowerShell 导入模块。
 - 运行导入 cmdlet，以使用 XML 文件作为输入导入策略和/或场数据（应用程序）。
- 完成迁移后的步骤。

运行实际迁移之前，可以先导出 XenApp 6.x 设置，然后在 XenApp 7.6 站点执行预览导入。通过预览可以识别潜在的故障点，使您可以在实际运行导入之前解决问题。例如，预览可能会检测到新 XenApp 7.6 站点中已经存在同名的应用程序。您也可以将通过预览生成的日志文件作为迁移指南。

除非另有说明，否则术语 6.x 是指 XenApp 6.0 或 6.5。

2014 年 12 月版本（版本 20141125）包含下列更新：

- 如果在 XenApp 6.x 场中使用迁移工具时遇到问题，请将问题报告到支持论坛 <http://discussions.citrix.com/forum/1411-xenapp-7x/>，以便 Citrix 能够调查这些问题，进而发现工具的潜在可改进功能。
- 新软件包 XAMigration.zip 文件现在包含两个单独的独立软件包：ReadIMA.zip 和 ImportFMA.zip。要从 XenApp 6.x 服务器导出，只需使用 ReadIMA.zip。要导入到 XenApp 7.6 服务器，只需使用 ImportFMA.zip。
- Export-XAFarm cmdlet 支持新参数 (EmbedIconData)，从而无需再将图标数据复制到单独的文件中。
- Import-XAFarm cmdlet 支持三个新参数。
 - MatchServer - 从名称与表达式匹配的服务器导入应用程序
 - NotMatchServer - 从名称与表达式不匹配的服务器导入应用程序
 - IncludeDisabledApps - 导入已禁用的应用程序
- 不导入预启动应用程序。
- Export-Policy cmdlet 适用于 XenDesktop 7.x。

迁移工具在 XenApp 7.6 Citrix [下载站点](#)下提供。XAMigration.zip 文件包含两个单独的独立软件包：

- ReadIMA.zip - 包含用于从 XenApp 6.x 场导出数据的数据文件，以及一些共享模块。

模块或文件	说明
ExportPolicy.psm1	用于将 XenApp 6.x 策略导出至 XML 文件的 PowerShell 脚本模块。
ExportXAFarm.psm1	用于将 XenApp 6.x 场设置导出至 XML 文件的 PowerShell 脚本模块。
ExportPolicy.psd1	ExportPolicy.psm1 脚本模块的 PowerShell 清单文件。

模块或文件	说明
ExportXAFarm.psd1	ExportXAFarm.psm1 脚本模块的 PowerShell 清单文件。
LogUtilities.psm1	包含日志记录功能的共享 PowerShell 脚本模块。
XmlUtilities.psd1	XmlUtilities.psm1 脚本模块的 PowerShell 清单文件。
XmlUtilities.psm1	包含 XML 功能的共享 PowerShell 脚本模块。

- ImportFMA.zip - 包含用于将数据导入到 XenApp 7.6 场的文件，以及一些共享模块。

模块或文件	说明
ImportPolicy.psm1	用于将策略导入 XenApp 7.6 的 PowerShell 脚本模块。
ImportXAFarm.psm1	用于将应用程序导入 XenApp 7.6 的 PowerShell 脚本模块。
ImportPolicy.psd1	ImportPolicy.psm1 脚本模块的 PowerShell 清单文件。
ImportXAFarm.psd1	ImportXAFarm.psm1 脚本模块的 PowerShell 清单文件。
PolicyData.xsd	策略数据的 XML 架构。
XAFarmData.xsd	XenApp 场数据的 XML 架构。
LogUtilities.psm1	包含日志记录功能的共享 PowerShell 脚本模块。
XmlUtilities.psd1	XmlUtilities.psm1 脚本模块的 PowerShell 清单文件。
XmlUtilities.psm1	包含 XML 功能的共享 PowerShell 脚本模块。

- 并非导入所有的策略设置；请参阅[未导入的策略设置](#)。不受支持的设置将被忽略并记录到日志文件中。
- 尽管在导出操作过程中会将所有应用程序详细信息收集到输出 XML 文件中，但是仅将服务器安装的应用程序导入到 XenApp 7.6 站点。不支持已发布桌面、内容和大多数流应用程序（请参阅[分步说明：导入数据](#)中的 Import-XAFarm cmdlet 参数以了解例外情况）。
- 不会导入应用程序服务器。
- 许多应用程序属性均不会被导入，因为 XenApp 6.x Independent Management Architecture (IMA) 和 XenApp 7.6 FlexCast Management Architecture (FMA) 技术之间存在差异；请参阅[应用程序属性映射](#)。
- 导入过程中会创建交付组。有关使用参数过滤导入内容的详细信息，请参阅[高级用法](#)。
- 仅导入使用 AppCenter 管理控制台创建的 Citrix 策略设置；不导入使用 Windows 组策略对象 (GPO) 创建的 Citrix 策略设置。
- 迁移脚本仅用于从 XenApp 6.x 到 XenApp 7.6 的迁移。
- 深度超过五级的嵌入式文件夹不受 Studio 支持，并且不会被导入。如果您的应用程序文件夹结构中包含深度超过五级的文件夹，请考虑在导入之前减少嵌入式文件夹级别的数量。

导出脚本所创建的 XML 文件可以包含有关您的环境和组织的敏感信息，例如，用户名、服务器名及其他 XenApp 场、应用程序和策略配置数据。在安全环境中存储和处理这些文件。

使用 XML 文件作为输入来导入策略和应用程序之前，请认真查看这些文件，以确保其中没有任何未经授权的修改。

策略对象分配（以前称为策略过滤器）用于控制策略的应用。导入策略之后，请认真查看每个策略的对象分配，以确保导入不会导致任何安全漏洞。导入后，可以对策略应用几组不同的用户、IP 地址或客户端名称。导入后，允许/拒绝设置可能具有不同的含义。

脚本提供详尽的日志记录，这些日志记录跟踪所有的 cmdlet 执行、有意义的消息、cmdlet 执行结果、警告和错误。

- 记录大多数 Citrix PowerShell cmdlet 的使用。记录用于创建新站点对象的导入脚本中的所有 PowerShell cmdlet。
- 记录脚本执行进度，包括要处理的对象。
- 记录影响流状态的主要操作，包括从命令行发出的流。
- 记录打印到控制台的所有消息，包括警告和错误。
- 每行都具有时间戳，精确到毫秒。

Citrix 建议在运行每个导出和导入 cmdlet 时指定日志文件。

如果不指定日志文件名，日志文件将存储在当前用户的主文件夹中（在 PowerShell \$HOME 变量中指定），前提是存在此文件夹。否则，日志文件存储在脚本的当前执行文件夹中。默认日志名为“XFarmYYYYMMDDHHmmSS-xxxxxx”，后六位数字组成一个随机数字。

默认情况下，会显示所有进度信息。要禁止显示这些信息，请在导出和导入 cmdlet 中指定 NoDetails 参数。

通常，脚本在遇到错误时停止执行。您可以在清除错误条件后重新运行 cmdlet。

未被视为错误的条件将记录到日志中；多数条件作为警告报告，脚本继续执行。例如，不受支持的应用程序类型将作为警告报告并且不会被导入。已存在于 XenApp 7.6 站点中的应用程序不会被导入。XenApp 7.6 中已弃用的策略设置不会被导入。

迁移脚本使用多个 PowerShell cmdlet，可能不会记录所有的潜在错误。有关其他日志记录覆盖范围，请使用 PowerShell 日志记录功能。例如，PowerShell 脚本记录打印到屏幕的所有内容。有关详细信息，请参阅 Start-Transcript 和 Stop-Transcript cmdlet 的帮助。

重要：请务必在开始迁移之前完整阅读本文。

您应该了解有关执行策略、模块、cmdlet 和脚本的 PowerShell 概念。虽然不需要具备全面的脚本编写专业知识，但也应该理解要执行的 cmdlet。请在执行 cmdlet 前，使用 Get-Help cmdlet 来查看每个迁移 cmdlet 的帮助信息。例如：

Get-Help -full Import-XAFarm

在命令行指定日志文件，并始终在运行 cmdlet 后查看日志文件。如果脚本失败，检查并修复日志文件中识别出的错误，然后重新运行 cmdlet。

须知事项：

- 同时运行两个部署（XenApp 6.x 场和新的 XenApp 7.6 站点）时，为方便应用程序交付，可以在 StoreFront 或 Web Interface 中聚合这两个部署。请参阅有关您的 StoreFront 或 Web Interface 版本的 eDocs 文档（“管理”>“创建应用商店”）。
- 应用程序图标数据采用以下两种方法之一处理：
 - 如果在 Export-XAFarm cmdlet 中指定 EmbedIconData 参数，导出的应用程序图标数据将嵌入到输出 XML 文件中。
 - 如果不在 Export-XAFarm cmdlet 中指定 EmbedIconData 参数，导出的应用程序图标数据将存储在以输出 XML 文件的基础名称后附加字符串“-icons”的名称命名的文件夹下面。例如，如果 XmlOutputFile 参数为“FarmData.xml”，则会创建“FarmData-icons”文件夹来存储应用程序图标。此文件夹中的图标数据文件为 .txt 文件，采用已发布应用程序的浏览器名称命名（虽然这些文件是 .txt 文件，但存储的数据被编码为二进制图标数据，导入脚本可以读取这些数据以重新创建应用程序图标）。导入操作过程中，如果在导入 XML 文件所在的位置找不到图标文件夹，将为导入的每个应用程序使用常规图标。
- 脚本模块、清单文件、共享模块和 cmdlet 的名称相似。使用 Tab 键自动补齐功能时请小心操作以避免出错。例如，Export-XAFarm 是一个 cmdlet。ExportXAFarmpsd1 和 ExportXAFarmpsm1 是无法执行的文件。
- 在下面的分步说明部分中，大多数 <字符串> 参数值用引号括起来。这些是单字符串可选项。

从 XenApp 6.x 服务器导出：

- 导出必须在配置了控制器和会话主机（通常称为控制器）服务器模式的 XenApp 6.x 服务器上运行。
- 要运行导出 cmdlet，您必须是具有对象读取权限的 XenApp 管理员。您还必须具有足够的 Windows 权限，以便可以运行 PowerShell 脚本；下面的分步过程包含相关说明。
- 开始导出前，确保 XenApp 6.x 场处于正常状态。备份场数据库。使用 Citrix IMA Helper 实用程序 (CTX133983) 确认场的完整性：从“IMA 数据存储”选项卡，运行主检查（然后使用 DSCheck 选项解析有效的条目）。在迁移前修复问题有助于预防导出失败。例如，如果错误地从场删除了服务器，其数据可能仍保留在数据库中，这可能导致导出脚本中的 cmdlet 失败（例如，Get-XAServer-ZoneName）。如果 cmdlet 失败，脚本便会失败。
- 可以在具有活动用户连接的活动场上运行导出 cmdlet；导出脚本仅读取静态场配置和策略数据。

导入到 XenApp 7.6 服务器：

- 可以将数据导入到 XenApp 7.6 部署（以及支持的更高版本）。必须先安装 XenApp 7.6 控制器和 Studio 并创建站点，然后再导入从 XenApp 6.x 场导出的数据。虽然导入设置不需要 VDA，但通过 VDA 可以使应用程序文件类型变为可用。
- 要运行导入 cmdlet，您必须是具有对象读取和创建权限的 XenApp 管理员。完全权限管理员具有这些权限。您还必须具有足够的 Windows 权限，以便可以运行 PowerShell 脚本；下面的分步过程包含相关说明。
- 导入过程中不得存在其他活动的用户连接。导入脚本会创建多个新对象，如果其他用户在同一时间更改了配置，可能会出现中断。

您可以导出数据，然后使用带有 -Preview 参数的 cmdlets 来查看实际导入过程中将要发生的情况，而无需实际执行导入操作。日志将指出实际导入过程中发生的情况；如果出现错误，可以在实际开始导入之前解决这些问题。

此处提供了展示导出完整过程的视频。

要将数据从 XenApp 6.x 控制器导出到 XML 文件，请完成以下步骤。

1. 从 Citrix 下载站点下载 XAMigration.zip 迁移工具包。方便起见，请将其放在 XenApp 6.x 场和 XenApp 7.6 站点均可以访问的网络文件共享位置。在网络文件共享位置解压 XAMigration.zip。解压后应包括两个 zip 文件：ReadIMA.zip 和 ImportFMA.zip。
2. 以至少具有只读权限和运行 PowerShell 脚本的 Windows 权限的 XenApp 管理员身份登录到 XenApp 6.x 控制器。
3. 将 ReadIMA.zip 从网络文件共享复制到 XenApp 6.x 控制器。在控制器上将 ReadIMA.zip 解压并提取到一个文件夹中（例如：C:\XAMigration）。
4. 打开 PowerShell 控制台，将当前目录设置为脚本位置。例如：
cd C:\XAMigration
5. 通过运行 Get-ExecutionPolicy 检查脚本执行策略。
6. 将脚本执行策略至少设置为 RemoteSigned 以允许执行脚本。例如：
Set-ExecutionPolicy RemoteSigned
7. 导入模块定义文件 ExportPolicy.psd1 和 ExportXAFarm.psd1：
Import-Module .\ExportPolicy.psd1

Import-Module .\ExportXAFarm.psd1

须知事项：

- 如果打算仅导出策略数据，可以只导入 ExportPolicy.psd1 模块定义文件。同样，如果打算仅导出场数据，则只需导入 ExportXAFarm.psd1。
 - 导入模块定义文件还会添加所需的 PowerShell 管理单元。
 - 请勿导入 .psm1 脚本文件。
8. 要导出策略数据，应运行 Export-Policy cmdlet。

参数	说明
- XmlOutputFile "<字符串 >.xml"	XML 输出文件名；此文件将存储导出数据。必须包含 .xml 扩展名。此文件不得存在，但是如果指定了路径，父路径必须存在。 默认值：无；此为必需的参数。
-LogFile "<字符串>"	日志文件名。扩展名为可选。如果不存在此文件，将会创建。如果此文件存在并且也指定了 NoClobber 参数，将会生成错误；否则将覆盖文件的内容。 默认：请参阅 日志记录和错误处理
-NoLog	不生成日志输出。如果也指定了 LogFile 参数，此参数会覆盖 LogFile 参数。 默认值：False；生成日志输出
-NoClobber	不覆盖 LogFile 参数中指定的现有日志文件。如果日志文件不存在，此参数无效。 默认值：False；覆盖现有日志文件
-NoDetails	不向控制台发送关于脚本执行情况的详细报告。 默认值：False；向控制台发送详细报告
- SuppressLogo	不向控制台打印消息“XenApp 6.x to XenApp/XenDesktop 7.6 Migration Tool Version #yyyyMMdd-hhmm#”（XenApp 6.x 至 XenApp/XenDesktop 7.6 迁移工具版本 #yyyyMMdd-hhmm#）。此消息标识脚本版本，在执行故障排除时非常有用；因此，Citrix 建议省略此参数。 默认值：False；向控制台打印此消息

示例：以下 cmdlet 将策略信息导出到名为 MyPolicies.xml 的 XML 文件。操作记录到名为 MyPolicies.log 的文件中。

- Export-Policy -XmlOutputFile ".\MyPolicies.XML" -LogFile ".\MyPolicies.Log"
9. 要导出场数据，应运行 Export-XAFarm cmdlet，指定日志文件和 XML 文件。

参数	说明
-XmlOutputFile "<字符串>.xml"	XML 输出文件名；此文件将存储导出数据。必须包含 .xml 扩展名。此文件不得存在，但是如果指定了路径，父路径必须存在。 默认值：无；此为必需的参数。

参数	说明
-LogFile "<字符串>"	日志文件名。扩展名为可选。如果不存在此文件，将会创建。如果此文件存在并且也指定了 NoClobber 参数，将会生成错误；否则将覆盖文件的内容。 默认：请参阅日志记录和错误处理
-NoLog	不生成日志输出。如果也指定了 LogFile 参数，此参数会覆盖 LogFile 参数。 默认值：False；生成日志输出
-NoClobber	不覆盖 LogFile 参数中指定的现有日志文件。如果日志文件不存在，此参数无效。 默认值：False；覆盖现有日志文件
-NoDetails	不向控制台发送关于脚本执行情况的详细报告。 默认值：False；向控制台发送详细报告
-SuppressLogo	不向控制台打印消息“XenApp 6.x to XenApp/XenDesktop 7.6 Migration Tool Version #yyyyMMdd-hhmm#”（XenApp 6.x 至 XenApp/XenDesktop 7.6 迁移工具版本 #yyyyMMdd-hhmm#）。此消息标识脚本版本，在执行故障排除时非常有用；因此，Citrix 建议省略此参数。 默认值：False；向控制台打印此消息
-IgnoreAdmins	不导出管理员信息。有关用法信息，请参阅高级用法。 默认值：False；导出管理员信息
-IgnoreApps	不导出应用程序信息。有关用法信息，请参阅高级用法。 默认值：False；导出应用程序信息
-IgnoreServers	不导出服务器信息。 默认值：False；导出服务器信息
-IgnoreZones	不导出区域信息。 默认值：False；导出区域信息
-IgnoreOthers	不导出配置日志记录、负载评估程序、负载均衡策略、打印机驱动程序和工作组等信息。 默认值：False；导出其他信息 注意：-IgnoreOthers 开关的用途是允许您在存在不影响正在用于导出或导入的实际数据的错误时继续进行导出。
-ApplLimit <整数>	要导出的应用程序数。有关用法信息，请参阅高级用法。 默认值：导出所有应用程序
-EmbedIconData	将应用程序图标数据作为其他对象嵌入到同一 XML 文件中。 默认值：单独存储图标。有关详细信息，请参阅要求、准备和最佳实践。
-SkipApps <整数>	跳过的应用程序数。有关用法信息，请参阅高级用法。 默认值：不跳过任何应用程序

示例：以下 cmdlet 将场信息导出到名为 MyFarm.xml 的 XML 文件中。操作记录到 MyFarm.log 文件中。创建名为“MyFarm-icons”的文件夹，用于存储应用程序

图标数据文件；此文件夹与 MyFarmXML 位于相同位置。
Export-XAFarm -XmlOutputFile ".\MyFarm.XML" -LogFile ".\MyFarm.Log"

导出脚本完成后，在命令行指定的 XML 文件包含策略和 XenApp 场数据。应用程序图标文件包含图标数据文件，日志文件指示导出过程中发生的情况。

[此处](#)提供了展示导入完整过程的视频。

您可以在执行实际导入之前运行预览导入（发布带有 Preview 参数的 Import-Policy 或 Import-XAFarm cmdlet）并查看日志文件。

要使用导出时生成的 XML 文件将数据导入到 XenApp 7.6 站点，请完成以下步骤。

1. 以具有读写权限和运行 PowerShell 脚本的 Windows 权限的管理员身份登录到 XenApp 7.6 控制器。
2. 如果没有在网络文件共享上解压迁移工具包 XAMigration，请在此时解压。将 ImportFMA.zip 从网络文件共享复制到 XenApp 7.6 控制器。在控制器上将 ImportFMA.zip 解压并提取到一个文件夹中（例如：C:\XAMigration）。
3. 将 XML 文件（导出过程中生成的输出文件）从 XenApp 6.x 控制器复制到 XenApp 7.6 控制器上提取 ImportFMA.zip 文件的位置。
如果在运行 Export-XAFarm cmdlet 时选择不将应用程序图标数据嵌入到 XML 输出文件中，请确保将图标数据文件夹和文件复制到 XenApp 7.6 控制器上与输出 XML 文件相同的位置，此位置包含应用程序数据和提取的 ImportFMA.zip 文件。

4. 打开 PowerShell 控制台，将当前目录设置为脚本位置。

```
cd C:\XAMigration
```

5. 通过运行 Get-ExecutionPolicy 检查脚本执行策略。

6. 将脚本执行策略至少设置为 RemoteSigned 以允许执行脚本。例如：

```
Set-ExecutionPolicy RemoteSigned
```

7. 导入 PowerShell 模块定义文件 ImportPolicy.psd1 和 ImportXAFarm.psd1：

```
Import-Module .\ImportPolicy.psd1
```

```
Import-Module .\ImportXAFarm.psd1
```

须知事项：

- 如果打算仅导入策略数据，可以只导入 ImportPolicy.psd1 模块定义文件。同样，如果打算仅导入场数据，则只需导入 ImportXAFarm.psd1。
 - 导入模块定义文件还会添加所需的 PowerShell 管理单元。
 - 请勿导入 .psm1 脚本文件。
8. 要导入策略数据，请运行 Import-Policy cmdlet，并指定包含已导出的策略数据的 XML 文件。

参数	说明
-XmlInputFile "<字符串>.xml"	XML 输入文件名；此文件包含通过运行 Export-Policy cmdlet 收集到的数据。必须包含 .xml 扩展名。 默认值：无；此为必需的参数。
-XsdFile "<字符串>"	XSD 文件名。导入脚本使用此文件验证 XML 输入文件的语法。有关用法信息，请参阅 高级用法 。 默认值：PolicyData.XSD
-LogFile "<字符串>"	日志文件名。如果已将导出日志文件复制到此服务器，请考虑为导入 cmdlet 使用不同的日志文件名。 默认：请参阅 日志记录和错误处理
-NoLog	不生成日志输出。如果也指定了 LogFile 参数，此参数会覆盖 LogFile 参数。 默认值：False；生成日志输出
-NoClobber	不覆盖 LogFile 参数中指定的现有日志文件。如果日志文件不存在，此参数无效。 默认值：False；覆盖现有日志文件
-NoDetails	不向控制台发送关于脚本执行情况的详细报告。 默认值：False；向控制台发送详细报告
- SuppressLogo	不向控制台打印消息“XenApp 6.x to XenApp/XenDesktop 7.6 Migration Tool Version #yyyyMMdd-hhmm#”（XenApp 6.x 至 XenApp/XenDesktop 7.6 迁移工具版本 #yyyyMMdd-hhmm#）。此消息标识脚本版本，在执行故障排除时非常有用；因此，Citrix 建议省

参数	说明
	默认值：False；向控制台打印此消息
-Preview	执行预览导入：从 XML 输入文件读取数据，但不向站点导入对象。日志文件和控制台指示预览导入过程中发生的情况。预览向管理员显示实际导入过程中发生的情况。 默认值：False；发生实际导入

示例：以下 cmdlet 从名为 MyPolicies.xml 的 XML 文件导入策略数据。操作记录到名为 MyPolicies.log 的文件中。

```
Import-Policy -XmlInputFile ".\MyPolicies.XML" -LogFile ".\MyPolicies.Log"
```

9. 要导入应用程序，应运行 Import-XAFarm cmdlet，指定包含已导出的场数据的日志文件和 XML 文件。

参数	说明
-XmlInputFile "<字符串>.xml"	XML 输入文件名；此文件包含通过运行 Export-XAFarm cmdlet 收集到的数据。必须包含 .xml 扩展名。 默认值：无；此为必需的参数。
-XsdFile "<字符串>"	XSD 文件名。导入脚本使用此文件验证 XML 输入文件的语法。有关用法信息，请参阅 高级用法 。 默认值：XAFarmData.XSD
-LogFile "<字符串>"	日志文件名。如果已将导出日志文件复制到此服务器，请考虑为导入 cmdlet 使用不同的日志文件名。 默认：请参阅 日志记录和错误处理
-NoLog	不生成日志输出。如果也指定了 LogFile 参数，此参数会覆盖 LogFile 参数。 默认值：False；生成日志输出
-NoClobber	不覆盖 LogFile 参数中指定的现有日志文件。如果日志文件不存在，此参数无效。 默认值：False；覆盖现有日志文件
-NoDetails	不向控制台发送关于脚本执行情况的详细报告。 默认值：False；向控制台发送详细报告
-SuppressLogo	不向控制台打印消息“XenApp 6.x to XenApp/XenDesktop 7.6 Migration Tool Version #yyyyMMdd-hhmm#”（XenApp 6.x 至 XenApp/XenDesktop 7.6 迁移工具版本 #yyyyMMdd-hhmm#）。此消息标识脚本版本，在执行故障排除时非常有用；因此，Citrix 建议省略此参数。 默认值：False；向控制台打印此消息
-Preview	执行预览导入：从 XML 输入文件读取数据，但不向站点导入对象。日志文件和控制台指示预览导入过程中发生的情况。预览向管理员显示实际导入过程中发生的情况。 默认值：False；发生实际导入
-DeliveryGroupName "<字符串>"	所有导入应用程序的交付组名称。有关用法信息，请参阅 高级用法 。 默认值："<xenapp 场名称> - 交付组"
-MatchFolder "<字符串>"	仅导入名称与此字符串匹配的文件夹中的应用程序。有关用法信息，请参阅 高级用法 。 默认值：不进行匹配
-NotMatchFolder "<字符串>"	仅导入名称与此字符串不匹配的文件夹中的应用程序。有关用法信息，请参阅 高级用法 。

字符串>" 参数	默认值：不进行匹配 说明
-MatchServer "<字符串>"	仅从名称与此字符串匹配的服务器导入应用程序。有关用法信息，请参阅 高级用法 。
-NotMatchServer "<字符串>"	仅从名称与此字符串不匹配的服务器导入应用程序。有关用法信息，请参阅 高级用法 。 默认值：不进行匹配
-MatchWorkerGroup "<字符串>"	仅导入发布到名称与此字符串匹配的工作组中的应用程序。有关用法信息，请参阅 高级用法 。 默认值：不进行匹配
-NotMatchWorkerGroup "<字符串>"	仅导入发布到名称与此字符串不匹配的工作组中的应用程序。有关用法信息，请参阅 高级用法 。 默认值：不进行匹配
-MatchAccount "<字符串>"	仅导入发布到名称与此字符串匹配的用户帐户中的应用程序。有关用法信息，请参阅 高级用法 。 默认值：不进行匹配
-NotMatchAccount "<字符串>"	仅导入发布到名称与此字符串不匹配的用户帐户中的应用程序。有关用法信息，请参阅 高级用法 。 默认值：不进行匹配
-IncludeStreamedApps	导入“StreamedToClientOrServerInstalled”类型的应用程序。（不导入其他流应用程序。） 默认值：不导入流应用程序
-IncludeDisabledApps	导入已标记为禁用的应用程序。 默认值：不导入已禁用的应用程序

示例：以下 cmdlet 从名为 MyFarm.xml 的 XML 文件导入应用程序。操作记录到名为 MyFarm.log 的文件中。

```
Import-XAFarm -XmlInputFile ".\MyFarm.XML" -LogFile ".\MyFarm.Log"
```

10. 导入成功完成后，完成迁移后任务。

成功将 XenApp 6.x 策略和场设置导入到 XenApp 7.6 站点中时，请使用以下指南确保数据已正确导入。

● 策略和策略设置

导入策略实际上是一种复制操作，已弃用的设置和策略除外，这些设置和策略不会被导入。迁移后检查本质上是将两个站点作比较。

1. 日志文件列出导入和忽略的所有策略和设置。首先，检查日志文件并识别没有导入的设置和策略。
2. 比较 XenApp 6.x 策略和导入到 XenApp 7.6 的策略。设置的值应保持不变（已弃用的策略设置除外，如下一步骤所述）。
 - 如果策略的数量很小，可以采用并列视图的方式比较 XenApp 6.x AppCenter 中显示的策略和 XenApp 7.6 Studio 中显示的策略。
 - 如果策略的数量很大，通过视觉观察进行比较可能不太可行。在这种情况下，使用策略导出 cmdlet (Export-Policy) 将 XenApp 7.6 策略导出到其他 XML 文件，然后使用文本比较工具（如 windiff）将该文件的数据与从 XenApp 6.x 导出策略时使用的 XML 文件中的数据作比较。
3. 使用[未导入的策略设置](#)部分中的信息来确定导入过程中可能发生变化的内容。如果 XenApp 6.x 策略仅包含已弃用的设置，整个策略均不会被导入。例如，如果 XenApp 6.x 策略仅包含 HMR 测试设置，则会完全忽略此策略，因为 XenApp 7.6 中没有受支持的等效设置。有些 XenApp 6.x 策略设置不再受支持，但在 XenApp 7.6 中实现了等效的功能。例如，在 XenApp 7.6 中，可以通过编辑交付组为服务器操作系统计算机配置重新启动计划；此功能之前通过策略设置得以实现。
4. 检查并确认过滤器应用到 XenApp 7.6 站点的方式，并将其与在 XenApp 6.x 中的应用方式作比较；XenApp 6.x 场和 XenApp 7.6 站点的关键不同之处可能会改变过滤器的效果。

● 过滤器

仔细检查每个策略的过滤器。为确保过滤器在 XenApp 7.6 中的作用与最初在 XenApp 6.x 中的作用相同，可能需要进行必要的更改。

过滤器	注意事项
访问控制	访问控制应该包含与源 XenApp 6.x 过滤器相同的值，并且应该可以在不进行任何更改的情况下使用。
Citrix CloudBridge	简单的布尔值；应该可以在不进行任何更改的情况下即可使用。
客户端 IP 地址	列出客户端 IP 地址范围；每个范围应该被允许或拒绝。导入脚本保存这些值，但是，如果其他客户端连接到 XenApp 7.6 VDA 计算机，可能需要更改这些值。
客户端名称	与客户端 IP 地址过滤器类似，导入脚本保存这些值，但是，如果其他客户端连接到 XenApp 7.6 VDA 计算机，可能需要更改这些值。
组织单位	可能会保留这些值，具体取决于在导入 OU 时是否可以对其解析。请仔细检查此过滤器，尤其是在 XenApp 6.x 和 XenApp 7.6 计算机驻留在不同的域中时。如果没有正确配置过滤器值，策略可能会应用到错误的 OU 集。 OU 仅通过名称来表示，因此，将某个 OU 名称解析后，目标 OU 可能会与 XenApp 6.x 域中的 OU 包含不同的成员，但这种可能性很小。即使保留了 OU 过滤器的某些值，仍应仔细检查这些值。
用户或组	可能会保留这些值，具体取决于在导入帐户时是否可以对其解析。 与 OU 类似，帐户仅使用名称解析。因此，如果 XenApp 7.6 站点包含的域具有相同域名和用户名，但实际上是两个不同的域和用户，解析后的帐户可能不同于 XenApp 6.x 域用户。如果不正确检查和修改过滤器值，可能会出现错误的策略应用情况。
工作组	XenApp 7.6 不支持工作组。请考虑使用 XenApp 7.6 支持（XenApp 6.x 不支持）的交付组、交付组类型和标记过滤器。 - 交付组：允许基于交付组应用策略。每个过滤器条目指定一个交付组，可以允许或拒绝此交付组。 - 交付组类型：允许基于交付组类型应用策略。每个过滤器条目指定一个交付组类型，可以允许或拒绝此交付组类型。 - 标记：基于为 VDA 计算机创建的标记指定策略应用。可以允许或拒绝各个标记。

总而言之，如果 XenApp 6.x 场和 XenApp 7.6 站点位于不同的域中，需要重点关注一下涉及到域用户变更的过滤器。由于导入脚本在新域中仅使用域和用户名字符串来解析用户，因此可能会解析部分帐户，而另一部分帐户没有被解析。虽然不同的域和用户具有相同名称的可能性很小，但是仍应该仔细检查这些过滤器，以确保它们包含正确的值。

应用程序

应用程序导入脚本不仅导入应用程序，还创建对象，如交付组。如果应用程序导入涉及到多次迭代，原始应用程序文件夹层次结构可能会发生显著变化。

- 首先，读取迁移日志文件（其中包含导入了哪些应用程序、忽略了哪些应用程序的详细信息）和 cmdlet（用于创建应用程序）。
- 对于每个应用程序：
 - 通过视觉观察来确保导入过程中保留了基本属性。使用[应用程序属性映射](#)部分中的信息来确定哪些属性按原样导入、哪些没有被导入、哪些已经使用 XenApp 6.x 应用程序数据初始化。
 - 检查用户列表。导入脚本自动将用户的详细列表导入到 XenApp 7.6 中应用程序的限制可见性列表中。检查以确保此列表保持不变。
- 不会导入应用程序服务器。这意味着尚不可访问导入的所有应用程序。必须将包含这些应用程序的交付组分配到计算机目录，这些计算机目录包含具有已发布应用程序的可执行映像的计算机。对于每个应用程序：
 - 确保可执行文件名和工作目录指向存在于分配到交付组的计算机中的可执行文件（通过计算机目录）。
 - 检查命令行参数（可能是任何内容，如文件名、环境变量或可执行文件名）。确认参数对分配到交付组的计算机目录中的所有计算机有效。

日志文件

日志文件是进行导出和导入时最重要的参考资源。这是为什么在默认情况下不能覆盖现有日志文件，并且默认日志文件名应该唯一的原因。

如“日志记录和错误处理”部分所述，如果选择通过 PowerShell Start-Transcript 和 Stop-Transcript cmdlet（记录输入和打印到控制台的所有内容）使用其他日志记录覆盖范围，该输出和日志文件将提供关于导入和导出活动的完整参考。

使用日志文件中的时间戳可以诊断某些问题。例如，如果导出或导入运行很长时间，则可以确定存在故障的数据库连接或解析用户帐户是否占用了大部分时间。

通过日志文件中记录的命令还可以了解读取或创建某些对象的方式。例如，要创建交付组，会执行多个命令，不仅创建交付组本身，还会创建其他对象，如允许将应用程序对象分配到交付组的访问策略规则。

日志文件还可以用于诊断失败的导出或导入。通常，日志文件的最后一行会指出导致失败的原因；日志文件中还会保存失败错误消息。与 XML 文件结合使用时，日志文件还可以用于确定失败所涉及的对象。

检查和测试迁移后，您可以：

- 通过在服务器上运行 7.6 安装程序，将 XenApp 6.5 工作服务器升级到最新的 Virtual Delivery Agents (VDA)，此操作会删除 XenApp 6.5 软件，然后自动安装最新

的 VDA。有关说明，请参阅[将 XenApp 6.5 工作进程升级至新的 VDA for Windows Server OS](#)。

对于 XenApp 6.0 工作服务器，必须手动从服务器卸载 XenApp 6.0 软件。然后，可以使用 7.6 安装程序安装最新的 VDA。无法使用 7.6 安装程序自动删除 XenApp 6.0 软件。

2. 从新 XenApp 站点中的 Studio，为升级后的工作服务器创建计算机目录（或编辑现有目录）。
3. 将升级后的计算机从计算机目录添加到包含这些 VDA for Windows Server OS 上安装的应用程序的交付组。

默认情况下，Export-Policy cmdlet 将所有策略数据导出到 XML 文件中。同样，Export-XAFarm 将所有场数据导出到 XML 文件中。您可以使用命令行参数更加精确地控制要导出和导入的内容。

- **导出部分应用程序** - 如果具有大量应用程序，并且您希望控制要导出到 XML 文件中的数量，请使用以下参数：

- AppLimit - 指定要导出的应用程序的数量。
- SkipApps - 指定在导出后续应用程序之前要跳过的应用程序的数量。

可以同时使用这两个参数，以便于管理的批次导出大量应用程序。例如，首次运行 Export-XAFarm 时，希望仅导出前 200 个应用程序，则可以在 AppLimit 参数中指定该值。

Export-XAFarm -XmlOutputFile "Apps1-200.xml" -AppLimit "200"

下次运行 Export-XAFarm 时，您希望导出后面的 100 个应用程序，则可以使用 SkipApps 参数忽略已经导出的应用程序（即前 200 个应用程序），并使用 AppLimit 参数导出后面的 100 个应用程序。

Export-XAFarm -XmlOutputFile "Apps201-300.xml" -AppLimit "100" -SkipApps "200"

- **不导出某些对象** - 可以忽略某些对象，从而无需导出这些对象，特别是不会被导入的对象；请参阅[未导入的策略设置](#)和[应用程序属性映射](#)。使用以下参数防止导出不需要的对象：

- IgnoreAdmins - 不导出管理员对象
- IgnoreServers - 不导出服务器对象
- IgnoreZones - 不导出区域对象
- IgnoreOthers - 不导出配置日志记录、负载评估程序、负载平衡策略、打印机驱动器和工作组对象
- IgnoreApps - 不导出应用程序；通过此参数可以将其他数据导出到 XML 输出文件，然后再次运行导出，从而将应用程序导出到其他 XML 输出文件。

也可以使用这些参数解决可能会导致导出失败的问题。例如，如果区域中存在损坏的服务器，区域导出可能会失败；但如果使用 IgnoreZones 参数，则可以继续导出其他对象。

- **交付组名称** - 如果不希望将所有应用程序放置在一个交付组中（例如，因为这些应用程序供不同的用户集合访问并且发布到不同的服务器集合中），则可以多次运行 Import-XAFarm，每次指定不同的应用程序和不同的交付组。尽管可以在迁移后使用 PowerShell cmdlet 将应用程序从一个交付组移动到另一个交付组，但有选择性地导入到唯一的交付组可以减少或省去之后再移动应用程序的繁琐。

1. 结合使用 DeliveryGroupName 参数和 Import-XAFarm cmdlet。脚本在指定交付组不存在时创建相应的交付组。
2. 结合使用以下参数和正则表达式基于文件夹、工作组、用户帐户和/或服务器名称来过滤要导入到交付组中的应用程序。建议在正则表达式两边使用单引号或双引号。有关正则表达式的信息，请参阅[http://msdn.microsoft.com/zh-cn/library/hs600312\(v=vs.110\).aspx](http://msdn.microsoft.com/zh-cn/library/hs600312(v=vs.110).aspx)。

- MatchWorkerGroup 和 NotMatchWorkerGroup - 例如，对于发布到工作组的应用程序，以下 cmdlet 将名为“Productivity Apps”的工作组中的应用程序导入到具有相同名称的 XenApp 7.6 交付组中。

Import-XAFarm -XmlInputFile XAFarm.xml -LogFile XAFarmImport.log -MatchWorkerGroup 'Productivity Apps' -DeliveryGroupName 'Productivity Apps'

- MatchFolder 和 NotMatchFolder - 例如，对于采用应用程序文件夹组织的应用程序，以下 cmdlet 将名为“Productivity Apps”文件夹中的应用程序导入到具有相同名称的 XenApp 7.6 交付组中。

Import-XAFarm -XmlInputFile XAFarm.xml -LogFile XAFarmImport.log -MatchFolder 'Productivity Apps' -DeliveryGroupName 'Productivity Apps'

例如，下列 cmdlet 将名称中包含“MS Office Apps”的任何文件夹中的应用程序导入到默认交付组中。

Import-XAFarm -XmlInputFile .\TheFarmApps.XML -MatchFolder ".*MS Office Apps.*"

- MatchAccount 和 NotMatchAccount - 例如，对于发布到 Active Directory 用户或用户组的应用程序，以下 cmdlet 将发布到用户组“Finance Group”的应用程序导入到名为“Finance”的 XenApp 7.6 交付组中。

Import-XAFarm -XmlInputFile XAFarm.xml -LogFile XAFarmImport.log -MatchAccount 'DOMAIN\Finance Group' -DeliveryGroupName 'Finance'

- MatchServer 和 NotMatchServer - 例如，对于服务器上组织的应用程序，下列 cmdlet 将不以“Current”命名的服务器关联的应用程序导入到名为“Legacy”的 XenApp 交付组中。

Import-XAFarm -XmlInputFile XAFarm.xml -LogFile XAFarmImport.log -NotMatchServer 'Current' -DeliveryGroupName 'Legacy'

- **自定义** - PowerShell 程序可以创建自己的工具。例如，您可以将导出脚本用作清单工具来跟踪 XenApp 6.x 场中的更改。您还可以修改 XSD 文件（或创建自己的 XSD 文件）以存储其他数据或 XML 文件中采用其他格式的数据。您可以随每个导入 cmdlet 指定非默认的 XSD 文件。

注意：尽管您可以修改脚本文件以满足特定迁移要求或高级迁移要求，但是仅支持未修改状态的脚本。Citrix 技术支持人员会建议还原到未修改的脚本以确定预期行为或在必要时提供支持。

- 如果要使用 PowerShell 2.0，并且已使用 Add-PSSnapIn cmdlet 添加 Citrix Group Policy PowerShell 提供程序管理单元或 Citrix Common Commands 管理单元，运行导出或导入 cmdlet 时，可能会出现错误消息“Object reference not set to an instance of an object”（未将对象引用设置为对象的实例）。此错误不会影响脚本的执行，可以将其忽略。
- 请避免在使用导出和导入脚本模块的控制台会话中添加或删除 Citrix Group Policy PowerShell 提供程序管理单元，因为这些脚本模块会自动添加此管理单元。如果单独添加或删除此管理单元，可能会看到以下其中一条错误：

- “A drive with the name 'LocalGpo' already exists.”（已存在名为“LocalGpo”的驱动器。）管理单元被添加两次时会出现此错误。管理单元尝试在驱动器 LocalGpo 已加载的情况下装载此驱动器，然后报告错误。
- “A parameter cannot be found that matches parameter name 'Controller'.”（找不到与参数名“Controller”匹配的参数。）当未添加管理单元，而脚本尝试装载驱动器时会出现此错误。脚本不知道管理单元已被删除。请关闭控制台并启动新会话。在新会话中，导入脚本模块；不要单独添加或删除管理单元。
- 导入模块时，如果在 .psd1 文件上单击鼠标右键并选择打开或使用 PowerShell 打开，PowerShell 控制台窗口将快速打开并关闭，直到您停止进程。为避免此错误，请直接输入完整的 PowerShell 脚本模块名称（例如，Import-Module .\ExportPolicy.psd1）。
- 如果在运行导出或导入时收到权限错误，请确保您具有对象读取权限（对于导出）或对象的读取和创建权限（对于导入）的 XenApp 管理员。必须具有运行 PowerShell 脚本的 Windows 权限。
- 如果导出失败，请通过在 XenApp 6.x 控制器服务器上运行 DSMMAINT 和 DSCHECK 实用程序来检查 XenApp 6.x 场是否处于正常状态。
- 如果运行预览导入，然后再次运行导入 cmdlet 以执行实际迁移，结果发现未导入任何内容，请验证是否从导入 cmdlet 删除 Preview 参数。

由于不再支持以下计算机和用户策略设置，因此不会导入这些策略设置。请注意，未过滤的策略永不导入。支持这些设置的功能和组件已由新技术/组件替代，或者由于架构和平台变化导致这些设置不再适用。

未导入的计算机策略设置

- 连接访问控制
- CPU 管理服务器级别
- DNS 地址解析
- 场名称
- 全图标缓存
- 运行状况监视、运行状况监视测试
- 许可证服务器主机名、许可证服务器端口
- 限制用户会话、管理员会话限制
- 负载评估程序名称
- 登录限制事件日志记录
- 具有登录限制的最大服务器百分比
- 内存优化、内存优化应用程序排除列表、内存优化时间间隔、内存优化计划: 月日期、内存优化计划: 周日期、内存优化计划: 时间
- 脱机应用程序客户端信任、脱机应用程序事件日志记录、脱机应用程序许可证期限、脱机应用程序用户
- 提示输入密码
- 重新启动自定义警告、重新启动自定义警告文本、重新启动登录禁止时间、重新启动计划频率、重新启动计划随机时间间隔、重新启动计划开始日期、重新启动计划时间、重新启动警告时间间隔、重新启动警告开始时间、重新启动对用户的警告、排定的重新启动
- 重影 *
- 信任 XML 请求（在 StoreFront 中配置）
- 虚拟 IP 适配器地址过滤、虚拟 IP 兼容性程序列表、虚拟 IP 增强兼容性、虚拟 IP 过滤器适配器地址程序列表
- 工作负载名称
- XenApp 产品版本、XenApp 产品模型
- XML Service 端口

* 由 Windows 远程协助取代

未导入的用户策略设置

- 自动连接客户端 COM 端口、自动连接客户端 LPT 端口
- 客户端 COM 端口重定向、客户端 LPT 端口重定向
- 客户端打印机名称
- 并发登录限制
- 从重影连接输入 *
- 延迟断开连接计时器间隔、延迟终止计时器间隔
- 记录重影尝试 *
- 通知用户有挂起的重影连接 *
- 预启动断开连接计时器间隔、预启动终止计时器间隔
- 会话重要性
- Single Sign-On、Single Sign-On 中央存储
- 可重影其他用户的用户、无法重影其他用户的用户 *

* 由 Windows 远程协助取代

不会导入以下应用程序类型。

- 服务器桌面
- 内容

- 流应用程序（App-V 是用于流应用程序的新方法）

场数据导入脚本仅导入应用程序。以下应用程序属性按原样导入。

IMA 属性	FMA 属性
AddToClientDesktop	ShortcutAddedToDesktop
AddToClientStartMenu	ShortcutAddedToStartMenu
ClientFolder	ClientFolder
CommandLineExecutable	CommandLineExecutable
CpuPriorityLevel	CpuPriorityLevel
说明	说明
DisplayName	PublishedName
已启用	已启用
StartMenuFolder	StartMenuFolder
WaitOnPrinterCreation	WaitForPrinterCreation
WorkingDirectory	WorkingDirectory
FolderPath	AdminFolderName

注意：IMA 和 FMA 对文件夹名称长度的限制不相同。在 IMA 中，文件夹名称限制是 256 个字符；FMA 限制是 64 个字符。导入时，会跳过文件夹路径中包含的文件夹名称超过 64 个字符的应用程序。此限制仅适用于文件夹路径中的文件夹名称；整个文件夹路径的长度可以大于此限制。为避免在导入时跳过某些应用程序，Citrix 建议在导入前检查应用程序文件夹名称的长度并根据需要将其缩短。

默认情况下，以下应用程序属性可能已初始化，也可能未初始化，或者设置为 XenApp 6.x 数据中提供的值：

FMA 属性	值
名称	初始化为完整路径名称，其中包含 IMA 属性 FolderPath 和 DisplayName，但是去掉前导字符串 "Applications\"。
ApplicationType	HostedOnDesktop
CommandLineArguments	使用 XenApp 6.x 命令行参数初始化
IconFromClient	未初始化；默认为 False
IconUid	初始化为使用 XenApp 6.x 图标数据创建的图标对象
SecureCmdLineArgumentsEnabled	未初始化；默认为 True
UserFilterEnabled	未初始化；默认为 False
UUID	只读，由 Controller 分配

FMA 属性	集初始化；默认为 True

以下应用程序属性将部分迁移：

IMA 属性	备注
FileTypes	只迁移存在于新 XenApp 站点上的文件类型。不存在于新站点上的文件类型将被忽略。文件类型只有在新站点上的文件类型更新之后才会导入。
IconData	如果已经为导出的应用程序提供图标数据，将创建新图标对象。
帐户	应用程序的用户帐户在交付组的用户列表和应用程序的用户列表之间分隔开。显式用户用于初始化应用程序的用户列表。此外，向交付组的用户列表中添加用户帐户所在域的“域用户”帐户。

不会导入下列 XenApp 6.x 属性：

IMA 属性	备注
ApplicationType	忽略。
HideWhenDisabled	忽略。
AccessSessionConditions	由交付组访问策略替代。
AccessSessionConditionsEnabled	由交付组访问策略替代。
ConnectionsThroughAccessGatewayAllowed	由交付组访问策略替代。
OtherConnectionsAllowed	由交付组访问策略替代。
AlternateProfiles	FMA 不支持流应用程序。
OfflineAccessAllowed	FMA 不支持流应用程序。
ProfileLocation	FMA 不支持流应用程序。
ProfileProgramArguments	FMA 不支持流应用程序。
ProfileProgramName	FMA 不支持流应用程序。
RunAsLeastPrivilegedUser	FMA 不支持流应用程序。
AnonymousConnectionsAllowed	FMA 使用其他技术支持未经身份验证（匿名）连接。
ApplicationId、SequenceNumber	IMA-unique 数据
AudioType	FMA 不支持高级客户端连接选项。
EncryptionLevel	在交付组中启用/禁用 SecureICA。
EncryptionRequired	在交付组中启用/禁用 SecureICA。

SslConnectionEnabled IMA 属性	FMA 使用其他 SSL 实现方法。 备注
ContentAddress	FMA 不支持已发布的内容。
ColorDepth	FMA 不支持高级窗口外观。
MaximizedOnStartup	FMA 不支持高级窗口外观。
TitleBarHidden	FMA 不支持高级窗口外观。
WindowsType	FMA 不支持高级窗口外观。
InstanceLimit	FMA 不支持应用程序限制。
MultipleInstancesPerUserAllowed	FMA 不支持应用程序限制。
LoadBalancingApplicationCheckEnabled	FMA 使用其他技术支持负载均衡。
PreLaunch	FMA 使用其他技术支持会话预启动。
CachingOption	FMA 使用其他技术支持会话预启动。
ServerNames	FMA 使用其他技术。
WorkerGroupNames	FMA 不支持工作组。

迁移 XenDesktop 4

May 04, 2016

可以使用迁移工具将 XenDesktop 4 场的数据和设置传输到 XenDesktop 7.x 站点，此工具位于 XenDesktop 安装介质上的 Support > Tools > MigrationTool 文件夹中。此工具包括：

- 导出工具 XdExport，用于将 XenDesktop 4 场数据导出到 XML 文件（默认名称：XdSettings.xml）。XML 文件架构位于 XdFarmxsd 文件中。
- 导入工具 XdImport，用于通过运行 PowerShell 脚本 Import-XdSettings.ps1 导入数据。

要成功使用迁移工具，两个部署必须具有相同的虚拟机管理程序版本（例如 XenServer 6.2）和 Active Directory 环境。

不能使用此工具迁移 XenApp，也不能将 XenDesktop 4 迁移到 XenApp。

提示：可以将 XenDesktop 5（或更高 XenDesktop 版本）升级到当前 XenDesktop 版本；请参阅[升级部署](#)。

并非所有数据和设置都可以导出。以下配置项无法迁移，因为它们可以导出但不能导入：

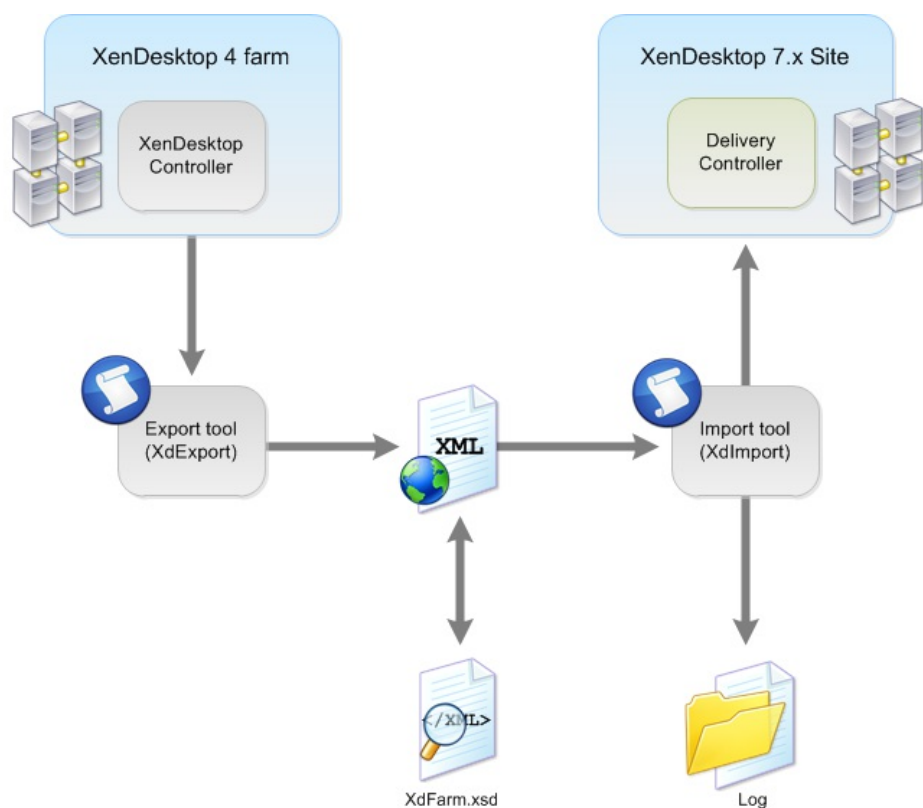
- 管理员
- 委派管理设置
- 桌面组文件夹
- 许可配置
- 注册表项

在迁移中不直接支持以下用例：

- 合并策略或桌面组的设置或托管设置。
- 将专用桌面合并到随机交付组。
- 通过迁移工具调整现有的组件设置。

有关详细信息，请参阅[迁移和不迁移的内容](#)。

下图总结了迁移过程。



迁移过程按照以下顺序进行：

1. 在 XenDesktop 4 控制器上的 Studio 控制台中，打开所有要导出的计算机的维护模式。
2. 使用 XdExport 将数据和设置从 XenDesktop 4 场导出到 XML 文件中；请参阅[从 XenDesktop 4 场中导出](#)。
3. 编辑 XML 文件，使其仅包含要导入到新 XenDesktop 站点中的数据和设置；请参阅[编辑迁移工具 XML 文件](#)。
4. 使用 XdImport 将数据和设置从 XML 文件导入到新 XenDesktop 站点；请参阅[导入 XenDesktop 4 数据](#)。
5. 要进行其他更改，请重复步骤 3 和 4。进行更改后，您可能需要将其他桌面导入到现有交付组。为此，请在导入时使用 Mergedesktops 参数。
6. 完成迁移后需执行的任务；请参阅[迁移后需执行的任务](#)。

在开始迁移之前，请完成以下操作：

- 确保您了解可以导出和导入哪些数据，以及如何应用于您自己的部署。请参阅[迁移和不迁移的内容](#)。
- Citrix 强烈建议您手动备份站点数据库，以便您可以在发现问题时进行还原。
- 安装 XenDesktop 7.x 组件并创建站点，包括数据库。
- 要从 XenDesktop 4 进行迁移，所有 VDA 必须处于 XenDesktop 5.x 级别，以便与 XenDesktop 4 和 XenDesktop 7.x 控制器兼容。Controller 基础结构全面运行 XenDesktop 7.x 后，Windows 7 VDA 可升级至 XenDesktop 7.x。有关详细信息，请参阅[迁移示例](#)。

从 XenDesktop 4 场中导出

May 04, 2016

导出工具 XdExport 可以从单个 XenDesktop 4 场中提取数据，并根据数据值的表示形式生成 XML 文件。

XML 文件的架构位于 XdFarm.xsd 文件中，该文件包含在下述迁移工具下载包中：XdExport.zip 和 XdImport.zip。

在要导出数据的场中的 XenDesktop 4 控制器上运行 XdExport。该计算机必须已安装 XenDesktop 4 PowerShell SDK。必须具有以下权限才可以导出数据：

- 用户身份至少是场的只读权限 Citrix 管理员。
- 读取注册表的权限。

您可以在 XenDesktop 控制器处于有效使用状态（例如，用户已登录 VDA）时运行该工具，但建议您不这样做。

Citrix 强烈建议：

- 在安装了公共修补程序的最新 XenDesktop 4 控制器上运行导出工具。
- 当导出运行时不更改站点的配置（例如，删除桌面组）。

1. 下载 XdExport.zip，并将文件提取到 XenDesktop 4 控制器中。
2. 在命令行提示窗口中，运行具有以下可选参数的 XdExport.exe：

参数	说明
-Verbose	生成提供详细进度信息的信息。
-FilePath <路径>	指示场数据要导出到的 XML 文件的位置。默认位置为 .\XdSettings.xml
- Overwrite	覆盖在 -Filepath 中指定的位置处存在的所有文件。如果未提供此参数，但已存在一个输出文件，此工具将无法运行，并显示以下错误消息：“错误: 文件已存在。请指定 -Overwrite 以允许覆盖该文件。”
-?或 -help	显示描述参数的文本，如果没有导出任何数据则退出。

3. 如果导出工具成功运行，则将显示完成消息。文件 XdSettings.xml 位于 FilePath 参数所指定的位置。如果导出工具运行失败，则将显示一条错误消息。

编辑迁移工具 XML 文件

May 04, 2016

将数据导入 XenDesktop 7.x 站点之前，请检查并编辑由导出工具 (XdExport) 生成的 XML 文件的内容（尤其是在分多个阶段执行迁移时），并导入部分用户、交付组和策略，然后再导入其他数据。

可以使用任意文本编辑器查看或更改文件内容，也可以使用专用 XML 编辑器，例如 Microsoft XML Notepad。

XML 内容中的某些元素必须存在，以便 XML 文件能被导入工具 (XdImport) 所接受。

所需的 XML 架构在 XdFarm.xsd 文件中定义，该文件作为迁移工具下载包的一部分提供。使用此文件时：

- 如果 minOccurs 属性的值为 1 或更大的值，则表示如果父元素存在，一些特定元素也必须存在。
- 如果提供给导入工具的 XML 文件无效，该工具将终止，并将显示一条错误消息，提示您如何找到该问题在 XML 文件中的位置。

要仅导入交付组和桌面的子集，可编辑 DesktopGroups 元素的内容。DesktopGroups 元素可以容纳多个 DesktopGroup 元素，并且在每个 DesktopGroup 元素中，都存在一个可以包含多个 Desktop 元素的 Desktops 元素。

请勿删除 DesktopGroups 元素，尽管您能够删除所有 DesktopGroup 元素并将其保留为空。同样，在每个 DesktopGroup 元素中，必须存在 Desktops 元素，但可以不包含任何 Desktop 元素。

删除 Desktop 或 DesktopGroup 元素以避免导入单个特定计算机或整个交付组。例如，XML 文件包含：

```
... ..
```

在此示例中，您的编辑操作将阻止导入 Group1 组，而只会导入 Group2 组中的 Machine3：

```
... ..
```

在 XenDesktop 4 中，桌面组可以通过文件夹形式组织，名称相同的桌面组可以显示在不同文件夹中，内部桌面组名称是显示给用户的名称。在此版本中，不可以将交付组置于文件夹中，每个交付组必须有唯一的内部名称，显示给用户的名称可以不同于内部名称。为反映这些不同，您可能必须重命名桌面组。

例如，在 XenDesktop 4 场中可以存在两个不同的桌面组，这两个桌面组显示给两个不同用户的名称均为“我的桌面”，这可以使用桌面组文件夹来实现。如果这些交付组要在 XenDesktop 7.x 站点中保持独立，您必须在 XML 文件中编辑桌面组名称，使这些名称为唯一的名称。

如果 XenDesktop 7.x 站点中的某个交付组与要导入的桌面组同名，并且这些交付组要在 XenDesktop 7.x 站点中保持独立，您必须在 XML 文件中编辑 XenDesktop 4 的桌面组名称，使该名称在站点中保持唯一。如果要导入的桌面组实际上与 XenDesktop 7.x 交付组相同，并且要将 XML 文件中的计算机合并到现有的桌面组中，这种情况下，无需重命名该交付组；为导入工具指定 -MergeDesktops 参数即可。例如，如果 XML 文件包含：

```
... \Sales ... \Finance
```

删除重复的名称，如下所示：

```
... \Sales ... \Finance
```

可以从 XML 文件中删除策略，并且可以指定唯一的名称以避免策略名称重复。不支持合并策略。

- 导入策略数据时有两种情形：成功导入所有策略，或者不导入任何策略数据（如有任何故障）。
- 导入大量带有许多设置的策略时，导入过程可能需要数小时的时间。
- 如果您批量导入策略，它们的原始优先级可能会受到影响。导入策略时，所导入策略的相对优先级将会保留，但这些策略的

优先级高于站点中已存在的策略。例如，如果要导入优先级分别为 1 到 4 的四个策略，并决定分两批导入这些策略，则应首先导入优先级为 3 和 4 的策略，因为第二批策略会自动获得更高的优先级。

要仅将策略的子集导入到 XenDesktop 7.x 站点中，请编辑 Policies 元素的内容。Policies 元素可以容纳多个 Policy 元素。请勿删除 Policies 元素，尽管您能够删除所有 Policy 元素并将其保留为空。删除全部 Policy 元素以避免导入特定的 XenDesktop 4 场策略。例如，如果 XML 文件包含：

... ..

为避免导入任何 XenDesktop 4 策略，并避免与已在 XenDesktop 7.x 站点中配置的策略相冲突，请按如下所示编辑该文件以删除各 Policy 元素：

或者，也可以按如下所示编辑该文件，以使用其他名称导入策略：

... ..

导入 XenDesktop 4 数据

May 04, 2016

导入工具 XdImport 从 XenDesktop 4 读取导出工具 XdExport 生成的 XML 文件中包含的设置，并将这些设置应用到现有 XenDesktop 7.x 站点。导入工具使用 PowerShell 脚本 Import-XdSettings.ps1。

要仅应用所导出数据子集，请先编辑此 XML 文件，然后再运行导入工具。例如，您可能希望删除在 XenDesktop 7.x 部署中不需要的桌面组和策略。如果您将全部元素留空，则导入工具可以成功运行。例如，您可以删除所有桌面组而不会出现任何问题。导入工具始终先验证 XML 文件，然后再尝试导入任何数据。

在已安装所有 XenDesktop 7.x SDK 的任意计算机上运行 XdImport。您必须具有完全权限 XenDesktop 管理员身份才能运行此工具。

在导入之前，请确保已设置 XenDesktop 7.x 站点，包括其数据库。Citrix 建议您先完成对 XenDesktop 7.x 的导入操作，然后再执行任何用户测试或常规站点配置。请仅在站点未使用时合并配置。

- 1. 创建一个 XenDesktop 7.x 站点。
- 2. 下载 XdImport.zip，并将文件提取到要运行导入工具的计算机中。
- 3. 在 PowerShell 会话中，运行带有以下参数的 Import-XdSettings.ps1：

参数	说明
- HypervisorConnectionCredentials	(必需。) PowerShell 哈希表，用于根据创建虚拟机管理程序连接的需要将虚拟机管理程序地址映射到 PSCredential 实例。默认值为 @{} 输入 XenDesktop 4 场连接的虚拟机管理程序的凭据。 对于单个虚拟机管理程序，可以按如下所示创建参数： \$credential = Get-Credential \$mappings = @"http://" + \$credential} .\Import-XdSettings.ps1 -FilePath .\XdSettings.xml -HypervisorConnectionCredentials \$mappir 在哈希表中指定的地址必须与 XML 文件中的地址完全一致。 例如，对于 XenServer 和 VMware 虚拟机管理程序，可以创建以下参数： \$Xencredential = Get-Credential \$VMWcredential = Get-Credential \$mappings = @"http://" + \$Xencredential;"http://SDK" = \$VMWcredential} .\Import-XdSettings.ps1 -FilePath .\XdSettings.xml -HypervisorConnectionCredentials \$mappings -AdminAddress localhost -MergeDesktops -SkipMachinePolicy -WhatIf -LogFilePath .\Import-XdSettings.log -?或 -help
-FilePath <路径>	(<路径> 的值为必需。) 要从中导入场数据的 XML 文件的位置。
-AdminAddress	XenDesktop 7.x 站点中的控制器名称。默认名称为 localhost
-MergeDesktops	将在 XML 文件中定义的桌面添加到 XenDesktop 7.x 站点中与该 XML 文件中所述的组同名的交付组中。同时还将添加相关联的计算机和用户。 如果未提供此参数，则不会将任何内容添加到 XenDesktop 7.x 站点的现有交付组中。
-SkipMachinePolicy	此脚本不创建包含站点级别设置的计算机策略。如果您未提供此参数，但站点的计算机策略存在，脚本运行将失败。
-WhatIf	完成试运行，以确定在 XenDesktop 7.x 站点中将发生哪些更改或添加哪些内容。如果包括此参数，则会向日志文件发送信息，但不会更改站点。
-LogFilePath <路径>	指明日志文件的完整路径。日志文件中包含针对 XenDesktop 7.x 站点执行的所有写操作的说明文本。默认值 = .\Import-XdSettings.log
-?或 -help	显示有关参数的信息，如果没有导入任何数据则退出。

如果 XML 文件中包含策略数据，则系统或者成功导入所有策略，或者不导入任何策略数据（如有任何故障）。导入大量带有许多设置的策略时，导入过程可能需要数小时的时间。

脚本完成时，将显示完成消息。从 XML 文件中成功导入数据后，可以进一步反复运行导出和导入过程，或者，如果已导入所有相关数据，则完成迁移后任务。

迁移后需执行的任务

May 04, 2016

从 XenDesktop 4 场成功将数据导入 XenDesktop 7.x 站点后，在使用新站点进行生产工作之前，请执行以下任务：

- 升级 Virtual Delivery Agent (VDA)。Citrix 建议您在升级控制器、Studio 或 Director 之前先升级 VDA，但这并非必需操作。
 - 对于 Windows Vista 和 Windows XP，请升级到 XenDesktop 5.6 Feature Pack 1 Virtual Desktop Agent。
 - 对于 Windows 7，请升级到 XenDesktop 7.x Virtual Delivery Agent。
- 为 XenDesktop 7.x 站点创建所需的管理员。
- 更新用户设备 — Citrix 建议使用最新版本的 Citrix Receiver 更新用户设备，以获取修补程序所带来的好处以及获取对最新功能的支持。
- 修改导入的桌面以使用基于注册表的控制器发现，并使用以下任一方法将其指向 XenDesktop 7.x 控制器：
 - 手动编辑注册表以删除不必要的组织单位 (OU) GUID 注册表项，并添加 ListOfDDCs 注册表项。
 - 设置一个计算机策略以通过使用 Active Directory 策略 GPMC.msc 将控制器列表分发给各个桌面。不能使用 Studio 配置此设置。

对于 XenDesktop 7.x 来说，基于注册表的 Controller 发现为其默认使用的发现方法，但仍然可以使用基于 Active Directory 的发现。

- 或者，还可以选择实施 [CTX133384](#) 的 XenDesktop 基于注册表的注册最佳做法中所述的以下注册表项设置：
 - HeartbeatPeriodMS
 - PrepareSessionConnectionTimeoutSec
 - MaxWorkers
 - DisableActiveSessionReconnect
 - ControllersGroupGuid

如果您不执行此操作，则使用这些注册表项的默认 XenDesktop 7.x 设置。

- 使导入的计算机脱离维护模式（如果这些计算机在 XenDesktop 4 中处于维护模式），然后再生成 XML 文件。
- 检查 XenDesktop 7.x 设置以确保其正确无误，特别是当您在 XenDesktop 4 中更改了 PortICAConfig XML 文件时。
- 检查所有已迁移的组件，确保迁移已成功。

迁移示例

May 04, 2016

在本例中，XenDesktop 4 场正在使用。XenDesktop 4 场有 50 个桌面组，每个组中平均包含 100 个桌面。XenDesktop 4 桌面是通过 Provisioning Services (PVS) 提供的，并且计算机在 VMware ESX 虚拟机管理程序上运行。所有 VM 上安装的 VDA 均为 XenDesktop 4 版本。

迁移步骤

1. 将所有 XenDesktop 4 VDA 升级至 XenDesktop 5.6 Feature Pack 1 VDA 软件。这样，VDA 便可以在 XenDesktop 4 控制器和 XenDesktop 7 Delivery Controller 中注册。
 - 对于 Windows 7 VDA，请参阅在 [VM 或刀片式计算机上升级 Virtual Desktop Agent](#)。
 - 对于 Windows XP 和 Windows Vista VDA，请参阅 [Windows XP 或 Windows Vista 上的 Virtual Desktop Agent](#)。
2. 确保所有用户均已从 XenDesktop 4 场注销。
3. 确保所有这些计算机均处于维护模式。
4. 在 XenDesktop 4 场中运行导出工具 (XdExport)。
5. 安装 XenDesktop 7 组件。
 1. 使用 Studio 创建全面生产模式站点。
 2. 如果 Provisioning Services 是部署的一部分，请升级 Provisioning Services 服务器和代理。
 3. 升级许可证服务器及关联的许可证。
6. 将导入工具 (XdImport) 解压缩到 XenDesktop 7 Controller 上的本地目录。
7. 将导出工具在步骤 4 中生成的 XML 文件 (XdSettings.xml) 复制到本地目录。
8. 从 XenDesktop 7 站点上 Studio 根节点的 PowerShell 控制台中，启动 PowerShell 会话。
9. 运行导入工具 (XdImport)，以传递关联虚拟机管理程序凭据和 XML 文件路径。
10. 从 Studio 导航窗格中的“管理员”节点重新手动创建管理员设置；有关详细信息，请参阅[委派管理](#)。
11. 修改导入的桌面，以使用基于注册表的 Controller 发现；并将其指向新的 XenDesktop 7 控制器。
12. 对于在 Windows 7 上运行的 VDA，Citrix 建议升级这些 VDA，以使用 XenDesktop 7 VDA for Windows Desktop OS，这样便可以访问所有新功能。

针对目录或交付组中的计算机将 VDA 升级到 XenDesktop 7 后，升级目录（请参阅[管理计算机目录](#)）和交付组（请参阅[管理交付组中设置](#)）。
13. 关闭交付组的维护模式。
14. 配置 StoreFront 以提供之前通过 Web Interface 提供的桌面。请参阅 StoreFront 文档。

在本例中，迁移分为多个步骤进行，每个步骤均会迁移部分剩余桌面。XenDesktop 4 场正在使用，XenDesktop 7.1 站点已创建并且正在使用。XenDesktop 4 场有 50 个桌面组，每个组中平均包含 100 个桌面。XenDesktop 4 桌面是通过 Provisioning Services 提供的，并且计算机在 Citrix XenServer 虚拟机管理程序上运行。所有 VM 上安装的 VDA 均为 XenDesktop 4 版本。

迁移步骤

1. 在 XenDesktop 4 场中运行导出工具。
 1. 在场中的一个 Desktop Delivery Controller 上解压缩导出工具 (XdExport)。
 2. 以 Citrix 管理员身份运行不具有参数的导出工具。
2. 复制并编辑生成的 XML 文件，以使其仅包含要迁移的组和桌面。
3. 在 XenDesktop 4 场中，确保要迁移的桌面上的所有用户均已注销，并打开要迁移的所有桌面的维护模式。

4. 将导入工具 (XdiImport) 解压缩到 XenDesktop 7.1 Delivery Controller 上的本地目录。
5. 将已编辑的 XML 复制到本地目录。
6. 从 XenDesktop 7.1 站点上 Studio 根节点的 PowerShell 控制台中，启动 PowerShell 会话。
7. 运行导入工具 (XdiImport)，以传递关联虚拟机管理程序凭据和 XML 文件路径。
8. 从 Studio 导航窗格中的“管理员”节点重新手动创建管理员设置；有关详细信息，请参阅[委派管理](#)。
9. 修改导入的桌面，以使用基于注册表的 Controller 发现；并将其指向新的 XenDesktop 7.1 控制器。
10. 将所有 VDA 升级至相应的 VDA 软件：
 - 对于 Windows 7 VDA：
 - 按在 [VM 或刀片式计算机上升级 Virtual Desktop Agent](#) 中所述升级到 XenDesktop 7 Virtual Delivery Agent。
 - 针对目录或交付组中的计算机将所有 VDA 软件升级到 XenDesktop 7 后，升级目录（请参阅[管理计算机目录](#)）和交付组（请参阅[管理交付组中设置](#)）。
 - 对于 Windows XP 和 Windows Vista VDA，请升级到 XenDesktop 5.6 FP1；请参阅 [Windows XP 或 Windows Vista 上的 Virtual Desktop Agent](#)。
11. 关闭交付组的维护模式。
12. 配置 StoreFront 以提供之前通过 Web Interface 提供的桌面。请参阅 StoreFront 文档。

迁移和不迁移的内容

May 04, 2016

下表的描述虽不全面，但介绍了迁移到此版本的过程中最重要的数据所发生的行为。除非另有说明，否则导入数据类型。

数据类型	备注
桌面组	在本版本中桌面组变为交付组。桌面组图标不导出。 如果在 XenDesktop 4 中 DefaultEncryptionLevel 未设置为 Basic，则 SecureIcaRequired 将设置为 True。 如果 XenDesktop 4 场中的一个桌面组与 XenDesktop 7.x 站点中的某个交付组同名，则可以将属于 XenDesktop 4 组的桌面添加到目标站点内的同名交付组中。 为此，请在运行导入工具时指定 MergeDesktops 参数。XenDesktop 4 组的设置并不覆盖 XenDesktop 7.x 交付组的设置。如果未指定此参数，但存在一个与 XML 文件中定义的组同名的组，则在导入任何数据之前，该工具将显示一条错误并停止。
桌面	无法将专用桌面添加到随机交付组中。无法将随机桌面添加到静态交付组中。
计算机	计算机导入到四个计算机目录中。导入工具在 XenDesktop 7.x 站点中自动创建以下计算机目录： • 已导入现有随机（针对池 VM） • 已导入现有静态（针对已分配的 VM） • 已导入物理随机（针对池 PC 或刀片式服务器） • 已导入物理静态（针对专有 PC 或刀片式服务器）。 后续任何计算机导入操作都将使用这四个计算机目录。
Pool Management 池	包括多池的池以及空闲池设置（包括时间安排）。 • 默认情况下，PeakBufferSizePercent 设置为 10%。 • 默认情况下，OffPeakBufferSizePercent 设置为 10%。 • 在本版本中，XenDesktop 4 的工作日设置中所有未选中的日期都将作为“周末电源时间方案”的组成部分导入。 • HostingXD4 操作时间向上舍入到最接近的分钟。 • 开始时间将向下舍入到最接近的小时。 • 结束时间将向上舍入到最接近的小时。
场设置	以下场设置将作为计算机策略导入： • IcaKeepAlive • AutoClientReconnect • SessionReliability 不导入用于启用 Flash 播放器的设置。

数据类型	备注工具将导入某些策略数据，过滤器、设置和打印机均作为用户策略导入。有关用户策略导出并导入的更多详细信息，请参阅本文档中的其他表格。 - 新访问策略规则将根据 XenDesktop 4 组设置进行创建。 - 导入策略时，将保留其相对优先级顺序。但在添加策略时，使用的优先级始终高于 XenDesktop 7.x 站点中任何现有策略的优先级。 - 不支持合并策略功能。 不存在用于将策略导入 Active Directory 中的选项。这些策略始终存储在站点中。
用户分配	
虚拟机管理程序设置	使用 XdImport 工具时，此参数为必需项。 导出工具将导出虚拟机管理程序地址，但不导出访问这些虚拟机管理程序所需的凭据。要在 XenDesktop 7.x 站点中创建虚拟机管理程序连接，必须从 XML 文件中提取地址，并创建一个 Powershell 哈希表，用于将这些地址映射到相关的凭据实例。然后在导入工具 HypervisorConnectionCredentials 参数中指定此哈希表。有关更多详细信息，请参阅导入 XenDesktop 4 数据 不支持合并或更新现有桌面组和虚拟机管理程序连接的虚拟机管理程序设置。
管理员	(不导入。) 导入工具不导入任何管理员数据，其中包括与委派的管理员有关的数据。需要为 XenDesktop 7.x 站点创建新的管理员。
许可配置	(不导入。) 包括诸如许可证服务器名称和版本等信息。不导出许可证文件。
桌面组文件夹	(不导入。) 本版本不支持桌面组文件夹。存在重复的桌面组名称（原因是 XenDesktop 4 场中不同的文件夹包含同名的组）时，如果您不在 XML 文件中编辑名称，导入工具将终止。
注册表项	(不导入。) 有关实现注册表项的信息，请参阅 迁移后需执行的任务 。

下表介绍了如何导出和导入用户策略数据。

XenDesktop 4 类别和设置	XML 文件	XenDesktop 7.x 类别和设置
带宽\视觉效果\会话限制 OEM 虚拟通道	ClientOEMVCBandwidth	不导入
客户端设备\资源\其他 关闭 OEM 虚拟通道	DisableOEMVirtualChannels	不导入

~~用户工作区中~~ 类别和设置 不使用客户端的本地时间	~~XML 文件~~ ClientLocalTime	~~导入~~ XenDesktop 7.x 类别和设置
安全性\加密 SecureICA 加密	ClientSecurityRequirement	不导入
带宽\SpeedScreen 使用有损压缩的图像加速	LossyCompression 设置	ICA\视频显示\静止图像 有损压缩级别 有损压缩阈值 超级压缩 ICA\视频显示\移动图像 渐进式压缩级别 渐进式压缩阈值
带宽\视觉效果 关闭桌面墙纸	TurnOffWallpaper	ICA\桌面 UI 桌面墙纸
带宽\视觉效果 菜单动画	TurnOffMenuWindowAnimation	ICA\桌面 UI 菜单动画
带宽\视觉效果 拖动时关闭窗口内容	DoNotShowWindowContentsWhileDragging	ICA\桌面 UI 拖动时查看窗口内容
带宽\视觉效果\会话限制 音频	ClientAudioBandwidth__AllowedBandWidth	ICA\带宽 音频重定向带宽限制
带宽\视觉效果\会话限制 剪贴板	ClientClipboardBandwidth__AllowedBandWidth	ICA\带宽 剪贴板重定向带宽限制
带宽\视觉效果\会话限制 COM 端口	ClientComBandwidth__AllowedBandWidth	COM 端口重定向功能在 XenDesktop 7.x 中已弃用
带宽\视觉效果\会话限制	ClientDriveBandwidth__AllowedBandWidth	ICA\带宽

驱动器 XenDesktop 4 类别和设置	XML 文件	文件重定向带宽限制 XenDesktop 7.x 类别和设置
带宽\视觉效果\会话限制 LPT 端口	ClientLptBandwidth__AllowedBandWidth	LPT 端口重定向功能在 XenDesktop 7.x 中已弃用
带宽\视觉效果\会话限制 整体会话	OverallBandwidth__AllowedBandWidth	ICA\带宽 总会话带宽限制
带宽\视觉效果\会话限制 打印机	LimitPrinterBandWidth__AllowedBandWidth	ICA\带宽 打印机重定向带宽限制
客户端设备\资源\音频 麦克风	ClientAudioMicrophone__TurnOn	ICA\音频 客户端麦克风重定向
客户端设备\资源\音频 音频质量	ClientAudioQuality__Quality	ICA\音频 音频质量
客户端设备\资源\音频 关闭扬声器	DisableClientAudioMapping	ICA\音频 客户端音频重定向
客户端设备\资源\驱动器 连接	ConnectClientDriveAtLogon__TurnOn	ICA\文件重定向 自动连接驱动器
客户端设备\资源\驱动器 关闭软盘驱动器	DisableClientDriveMapping__DisableFloppyDrive	ICA\文件重定向 客户端软盘驱动器
客户端设备\资源\驱动器 关闭硬盘驱动器	DisableClientDriveMapping__DisableHardDrive	ICA\文件重定向 客户端固定驱动器
客户端设备\资源\驱动器 关闭 CD-ROM 驱动器	DisableClientDriveMapping__DisableCdrom	ICA\文件重定向 客户端光盘驱动器
客户端设备\资源\驱动器 关闭远程驱动器	DisableClientDriveMapping__DisableRemote	ICA\文件重定向 客户端网络驱动器

XenDesktop 4 类别和设置 客户端设备\资源\驱动器 关闭 USB 磁盘驱动器	XML 文件 DisableClientDriveMapping__DisableUSB	XenDesktop 7.x 类别和设置 ICA\文件重定向 客户端可移动驱动器
客户端设备\资源\驱动器 \优化 异步写入	CDMAsyncWrites	ICA\文件重定向 用户异步写入
客户端设备\资源\其他 关闭剪贴板映射	DisableClientClipboardMapping	ICA 客户端剪贴板重定向
客户端设备\资源\端口 关闭 COM 端口	DisableClientCOMPortMapping	COM 端口重定向功能在 XenDesktop 7.x 中已弃用
客户端设备\资源\端口 关闭 LPT 端口	DisableClientLPTPortMapping	LPT 端口重定向功能在 XenDesktop 7.x 中已弃用
客户端设备\资源\USB USB	RemoteUSBDevices__DisableRemoteUSBDevices	ICA\USB 设备 客户端 USB 设备重定向
打印\客户端打印机 自动创建	ConnectClientPrinterAtLogon__Flag	ICA\打印\客户端打印机 自动创建客户端打印机
打印\客户端打印机 旧版客户端打印机	LegacyClientPrinters__TurnOn	ICA\打印\客户端打印机 客户端打印机名称
打印\客户端打印机 打印机属性保留	ModifiedPrinterProperties__WriteMethod	ICA\打印\客户端打印机 打印机属性保留
打印\客户端打印机 打印作业路由	ClientPrintingForNetworkPrinter__TurnOn	ICA\打印\客户端打印机 直接连接到打印服务器
打印\客户端打印机 关闭客户端打印机映射	DisableClientPrinterMapping	ICA\打印 客户端打印机重定向

打印\驱动程序 XenDesktop 4 类别和设置 本地打印机驱动程序自动安装	PrintDriverAutoInstall__TurnOn XML 文件	ICA\打印\驱动程序 XenDesktop 7.x 类别和设置 自动安装现成的打印机驱动程序
打印\驱动程序 通用驱动程序	ClientPrintDriverToUse	ICA\打印\驱动程序 通用打印驱动程序使用
打印\会话打印机 会话打印机	NetworkPrinters	ICA\打印 会话打印机
打印\会话打印机 选择客户端的默认打印机	DefaultToMainClientPrinter__NetworkDefault DefaultToMainClientPrinter__TurnOn	ICA\打印 默认打印机

此版本不支持所有的 XenDesktop 4 组件。将不迁移以下项目：

- **Virtual Delivery Agent** - 必须先将 VDA 升级到 XenDesktop 5.x 的最低版本，XenDesktop 7.x Delivery Controller 才能从 XenDesktop 4 管理虚拟桌面。有关升级 VDA 的详细信息，请参阅[迁移后需执行的任务](#)。
- **Controller** - 必须部署新的 Controller 服务器。无法将 XenDesktop 4 Controller 升级到 XenDesktop 7.x 站点。XenDesktop 7.x 站点无法加入 XenDesktop 4 场，XenDesktop 4 Controller 无法加入 XenDesktop 7.x 站点。另外，每个版本具有不同的服务器要求；XenDesktop 4 要求 Windows Server 2003；XenDesktop 7.x 要求更高的 Windows Server 版本。
- **Web Interface** - Citrix 建议在 XenDesktop 7.x 中使用 StoreFront。有关安装和设置详细信息，请参阅 StoreFront 文档。当 XenDesktop 安装程序检测到 Web Interface 时，它将安装 StoreFront，但不会删除 Web Interface。
- **Active Directory 组织单位 (OU) 配置** - 不支持在两个场或两个站点之间或者在场与站点之间共享组织单位 (OU)。如果您计划将新站点配置为使用基于 Active Directory 的控制器发现（而非默认的基于注册表的控制器发现），必须创建一个新的 OU，用以对其提供支持。
- **PortICAConfig XML 文件** - 如果更改了此文件的默认设置，可能需要通过组策略对象为新站点配置这些设置。
- 通过 XenDesktop 4 Service Pack 1 提供的配置日志记录设置。
- Provisioning Services 的相关数据。
- 应用程序。
- Controller 列表。
- NetScaler Gateway。
- 事件日志限制设置。

安全性

May 04, 2016

Citrix XenApp 和 XenDesktop 安全性入门

XenApp 和 XenDesktop 提供设计安全的解决方案，使您可以根据安全需求定制环境。

IT 面临着移动工作人员数据丢失或被盗的安全隐患。通过托管应用程序和桌面，XenApp 和 XenDesktop 将所有数据存储于数据中心内，从而安全地将敏感数据和知识产权与终端设备分开。当启用策略以允许数据传输时，所有数据均会加密。

XenDesktop 和 XenApp 数据中心还提供集中式监视服务和管理服务，从而更易于响应事件。Director 允许 IT 监视和分析可通过网络访问的数据，Studio 允许 IT 修补和修复数据中心内的大部分漏洞，而不是在每个最终用户设备本地解决问题。

XenApp 和 XenDesktop 还简化了审计和法规遵从性操作，因为调查人员可以使用集中化审核追踪来确定哪些人员访问了哪些应用程序和数据。Director 通过访问配置日志记录和 OData API 收集有关系统更新和用户数据使用情况的历史数据。

通过委派管理员功能，您可以设置管理员角色，从而在某一粒度级别控制对 XenDesktop 和 XenApp 的访问。这样一来，在您的组织内可以灵活地向某些管理员授予任务、操作和作用域的完全访问权限，而其他管理员仅具有有限的访问权限。

XenApp 和 XenDesktop 通过在不同的网络级别（从本地级别到组织单位级别）应用策略，向管理员提供对用户的粒度级控制。这种策略控制确定用户、设备或用户和设备组是否可以连接、复制/粘贴或映射本地驱动器，从而尽可能地降低对第三方临时工作人员的安全顾虑。管理员还可以使用 Desktop Lock 功能，因此，当阻止对最终用户设备的本地操作系统进行访问时，最终用户仅可以使用虚拟桌面。

管理员还可以通过将站点配置为针对 Controller 或在最终用户和 Virtual Delivery Agents (VDA) 之间使用安全套接字层 (SSL) 安全协议，来增加 XenApp 或 XenDesktop 的安全性。也可以在站点上启用传输层安全性 (TLS) 安全协议，从而为 TCP/IP 连接提供服务器身份验证、数据流加密和消息完整性检查功能。

XenApp 和 XenDesktop 还支持向 Windows 或特定应用程序提供多重身份验证。多重身份验证还可以用于管理 XenApp 和 XenDesktop 交付的所有资源。这些方法包括：

- 令牌
- 智能卡
- RADIUS
- Kerberos
- 生物识别

XenDesktop 可以与从身份管理到防病毒软件等多种第三方安全解决方案集成。<http://www.citrix.com/ready> 提供了支持的产品列表。

选择用于通用准则标准认证的 XenApp 和 XenDesktop 版本。有关这些标准的列表，请访问 <http://www.commoncriteriaportal.org/cc/>。

- [安全最佳实践和注意事项](#)
- [委派管理](#)
- [智能卡](#)
- [SSL](#)
- [Desktop Lock](#)

安全最佳实践和注意事项

Nov 02, 2016

This document describes:

- General security best practices when using this release, and any security-related differences between this release and a conventional computer environment
- Manage user accounts
- Manage user privileges
- Manage logon rights
- Configure user rights
- Configure service settings
- Deployment scenarios and their security implications
- Remote PC Access security considerations

Your organization may need to meet specific security standards to satisfy regulatory requirements. This document does not cover this subject, because such security standards change over time. For up-to-date information on security standards and Citrix products, consult <http://www.citrix.com/security/>.

Keep all machines in your environment up to date with security patches. One advantage is that you can use thin clients as terminals, which simplifies this task.

Protect all machines in your environment with antivirus software.

Protect all machines in your environment with perimeter firewalls, including at enclave boundaries as appropriate.

If you are migrating a conventional environment to this release, you may need to reposition an existing perimeter firewall or add new perimeter firewalls. For example, suppose there is a perimeter firewall between a conventional client and database server in the data center. When this release is used, that perimeter firewall must instead be placed so that the virtual desktop and user device are on one side, and the database servers and Delivery Controllers in the data center are on the other side. You should therefore consider creating an enclave within your data center to contain the database servers and Controllers. You should also consider having protection between the user device and the virtual desktop.

All machines in your environment should be protected by a personal firewall. When you install core components and Virtual Delivery Agents (VDAs), you can choose to have the ports required for component and feature communication opened automatically if the Windows Firewall Service is detected (even if the firewall is not enabled). You can also choose to configure those firewall ports manually. If you use a different firewall, you must configure the firewall manually.

Note: TCP ports 1494 and 2598 are used for ICA and CGP and are therefore likely to be open at firewalls so that users outside the data center can access them. Citrix recommends that you do not use these ports for anything else, to avoid the possibility of inadvertently leaving administrative interfaces open to attack. Ports 1494 and 2598 are officially registered with the Internet Assigned Number Authority (see <http://www.iana.org/>).

All network communications should be appropriately secured and encrypted to match your security policy. You can secure all communication between Microsoft Windows computers using IPsec; refer to your operating system documentation for details about how to do this. In addition, communication between user devices and desktops is secured through Citrix SecureICA, which is configured by default to 128-bit encryption. You can configure SecureICA when you are creating or updating an assignment; see [Change basic settings](#).

Manage user accounts

If the option to install App-V publishing components is selected when installing a Virtual Delivery Agent (VDA), or if this feature is added later, the local administrative account CtxAppVCOMAdmin is added to the VDA. If you use the App-V publishing feature, do not modify this account. If you do not need to use the App-V publishing feature, do not select it at installation time. If you later decide not to use the App-V publishing feature, you can disable or delete this account.

Grant users only the capabilities they require. Microsoft Windows privileges continue to be applied to desktops in the usual way: configure privileges through User Rights Assignment and group memberships through Group Policy. One advantage of this release is that it is possible to grant a user administrative rights to a desktop without also granting physical control over the computer on which the desktop is stored.

When planning for desktop privileges, note:

- By default, when non-privileged users connect to a desktop, they see the time zone of the system running the desktop instead of the time zone of their own user device. For information on how to allow users to see their local time when using desktops, see [Change basic settings](#).
- A user who is an administrator on a desktop has full control over that desktop. If a desktop is a pooled desktop rather than a dedicated desktop, the user must be trusted in respect of all other users of that desktop, including future users. All users of the desktop need to be aware of the potential permanent risk to their data security posed by this situation. This consideration does not apply to dedicated desktops, which have only a single user; that user should not be an administrator on any other desktop.
- A user who is an administrator on a desktop can generally install software on that desktop, including potentially malicious software. The user can also potentially monitor or control traffic on any network connected to the desktop.

Manage logon rights

Logon rights are required for both user accounts and computer accounts. As with Microsoft Windows privileges, logon rights continue to be applied to desktops in the usual way: configure logon rights through User Rights Assignment and group memberships through Group Policy.

The Windows logon rights are: log on locally, log on through Remote Desktop Services, log on over the network (access this computer from the network), log on as a batch job, and log on as a service.

For computer accounts, grant computers only the logon rights they require. The logon right "Access this computer from the network" is required:

- At VDAs, for the computer accounts of Delivery Controllers
- At Delivery Controllers, for the computer accounts of VDAs. See [Active Directory OU-based Controller discovery](#).
- At StoreFront servers, for the computer accounts of other servers in the same StoreFront server group

For user accounts, grant users only the logon rights they require.

According to Microsoft, by default the group Remote Desktop Users is granted the logon right "Allow log on through Remote Desktop Services" (except on domain controllers).

Your organization's security policy may state explicitly that this group should be removed from that logon right. Consider the following approach:

- The Virtual Delivery Agent (VDA) for Server OS uses Microsoft Remote Desktop Services. You can configure the Remote Desktop Users group as a restricted group, and control membership of the group via Active Directory group policies. Refer to Microsoft documentation for more information.
- For other components of XenApp and XenDesktop, including the VDA for Desktop OS, the group Remote Desktop Users is not required. So, for those components, the group Remote Desktop Users does not require the logon right "Allow log on through Remote Desktop Services"; you can remove it. Additionally:
 - If you administer those computers via Remote Desktop Services, ensure that all such administrators are already members of the Administrators group.
 - If you do not administer those computers via Remote Desktop Services, consider disabling Remote Desktop Services itself on those computers.

Although it is possible to add users and groups to the login right "Deny logon through Remote Desktop Services", the use of deny logon rights is not generally recommended. Refer to Microsoft documentation for more information.

Configure user rights

Delivery Controller installation creates the following Windows services:

- Citrix AD Identity Service (NT SERVICE\CitrixADIdentityService): Manages Microsoft Active Directory computer accounts for VMs.
- Citrix Analytics (NT SERVICE\CitrixAnalytics): Collects site configuration usage information for use by Citrix, if this collection been approved by the site administrator. It then submits this information to Citrix, to help improve the product.
- Citrix App Library (NT SERVICE\CitrixAppLibrary): Supports management and provisioning of AppDisks, AppDNA integration, and management of App-V.
- Citrix Broker Service (NT SERVICE\CitrixBrokerService): Selects the virtual desktops or applications that are available to users.
- Citrix Configuration Logging Service (NT SERVICE\CitrixConfigurationLogging): Records all configuration changes and other state changes made by administrators to the site.
- Citrix Configuration Service (NT SERVICE\CitrixConfigurationService): Site-wide repository for shared configuration.
- Citrix Delegated Administration Service (NT SERVICE\CitrixDelegatedAdmin): Manages the permissions granted to administrators.
- Citrix Environment Test Service (NT SERVICE\CitrixEnvTest): Manages self-tests of the other Delivery Controller services.
- Citrix Host Service (NT SERVICE\CitrixHostService): Stores information about the hypervisor infrastructures used in a XenApp or XenDesktop deployment, and also offers functionality used by the console to enumerate resources in a hypervisor pool.
- Citrix Machine Creation Service (NT SERVICE\CitrixMachineCreationService): Orchestrates the creation of desktop VMs.
- Citrix Monitor Service (NT SERVICE\CitrixMonitor): Collects metrics for XenApp or XenDesktop, stores historical information, and provides a query interface for troubleshooting and reporting tools.
- Citrix Storefront Service (NT SERVICE\CitrixStorefront): Supports management of StoreFront. (It is not part of the StoreFront component itself.)
- Citrix Storefront Privileged Administration Service (NT SERVICE\CitrixPrivilegedService): Supports privileged management operations of StoreFront. (It is not part of the StoreFront component itself.)

Delivery Controller installation also creates the following Windows services. These are also created when installed with other Citrix components:

- Citrix Diagnostic Facility COM Server (NT SERVICE\CdfSvc): Supports the collection of diagnostic information for use by Citrix Support.
- Citrix Telemetry Service (NT SERVICE\CitrixTelemetryService): Collects diagnostic information for analysis by Citrix, such that the analysis results and recommendations can be viewed by administrators to help diagnose issues with the site.

Except for the Citrix Storefront Privileged Administration Service, these services are granted the logon right Log on as a service and the privileges Adjust memory quotas for a process, Generate security audits, and Replace a process level token. You do not need to change these user rights. These privileges are not used by the Delivery Controller and are automatically disabled.

Configure service settings

Except for the Citrix Storefront Privileged Administration service and the Citrix Telemetry Service, the Delivery Controller Windows services listed above in the "Configure user rights" section are configured to log on as the NETWORK SERVICE identity. Do not alter these service settings.

The Citrix Storefront Privileged Administration service is configured to log on Local System (NT AUTHORITY\SYSTEM). This is required for Delivery Controller StoreFront operations that are not normally available to services (including creating Microsoft IIS sites). Do not alter its service settings.

The Citrix Telemetry Service is configured to log on as its own service-specific identity.

You can disable the Citrix Telemetry Service. Apart from this service, and services that are already disabled, do not disable any other of these Delivery Controller Windows services.

Your user environment can consist either of user devices that are unmanaged by your organization and completely under the control of the user, or of user devices that are managed and administered by your organization. The security considerations for these two environments are generally different.

- **Managed user devices** - Managed user devices are under administrative control; they are either under your own control, or the control of another organization that you trust. You may configure and supply user devices directly to users; alternatively, you may provide terminals on which a single desktop runs in full-screen-only mode. You should follow the general security best practices described above for all managed user devices. This release has the advantage that minimal software is required on a user device.

A managed user device can be set up to be used in full-screen-only mode or in window mode:

- If a user device is configured to be used in full-screen-only mode, users log on to it with the usual Log On To Windows screen. The same user credentials are then used to log on automatically to this release.
- If a user device is configured so that users see their desktop in a window, users first log on to the user device, then log on to this release through a Web site supplied with the release.
- **Unmanaged user devices** - User devices that are not managed and administered by a trusted organization cannot be assumed to be under administrative control. For example, you might permit users to obtain and configure their own devices, but users might not follow the general security best practices described above. This release has the advantage

that it is possible to deliver desktops securely to unmanaged user devices. These devices should still have basic antivirus protection that will defeat keylogger and similar input attacks.

- **Data storage considerations** - When using this release, you can prevent users from storing data on user devices that are under their physical control. However, you must still consider the implications of users storing data on desktops. It is not good practice for users to store data on desktops; data should be held on file servers, database servers, or other repositories where it can be appropriately protected.

Your desktop environment may consist of various types of desktops, such as pooled and dedicated desktops:

- Users should never store data on desktops that are shared amongst users, such as pooled desktops.
- If users store data on dedicated desktops, that data should be removed if the desktop is later made available to other users.
- **Mixed-version environments** Mixed-version environments are inevitable during some upgrades. Follow best-practice and minimize the time that Citrix components of different versions co-exist.

In mixed-version environments security policy, for example, may not be uniformly enforced.

Note: This is typical of other software products; the use of an earlier version of Active Directory only partially enforces Group Policy with later versions of Windows.

The following scenario describes a security issue that can occur in a specific mixed-version Citrix environment. When Citrix Receiver 1.7 is used to connect to a virtual desktop running the Virtual Delivery Agent in XenApp and XenDesktop 7.6 Feature Pack 2, the policy "Allow file transfer between desktop and client" is enabled in the Site but cannot be disabled by a Delivery Controller running XenApp and XenDesktop 7.1. It does not recognize the policy, which was released only in the later version of the product. This policy allows users to upload and download files to their virtual desktop – the security issue. To work around this, upgrade the Delivery Controller, or a standalone instance of Studio, to Version 7.6 Feature Pack 2 and then use GP to disable the policy. Alternatively, use local policy on all affected virtual desktops.

Remote PC Access implements the following security features:

- Smart card use is supported.
- When a remote session connects, the office PC's monitor appears as blank.
- Remote PC Access redirects all keyboard and mouse input to the remote session, except CTRL+ALT+DEL and USB-enabled smart cards and biometric devices.
- SmoothRoaming is supported for a single user only.
- When a user has a remote session connected to an office PC, only that user can resume local access of the office PC. To resume local access, the user presses Ctrl-Alt-Del on the local PC and then logs on with the same credentials used by the remote session. The user can also resume local access by inserting a smart card or leveraging biometrics, if your system has appropriate third-party Credential Provider integration.

This default behavior can be overridden by enabling Fast User Switching via Group Policy Objects (GPOs) or by editing the registry.

- By default, Remote PC Access supports automatic assignment of multiple users to a VDA. In XenDesktop 5.6 Feature Pack 1, administrators could override this behavior using the RemotePCAccess.ps1 PowerShell script. This release uses a registry entry to allow or prohibit multiple automatic remote PC assignments; this setting applies to the entire Site. Caution: Editing the registry incorrectly can cause serious problems that may require you to reinstall your operating system. Citrix cannot guarantee that problems resulting from the incorrect use of Registry Editor can be solved. Use Registry Editor at your own risk. Be sure to back up the registry before you edit it.

To restrict automatic assignments to a single user:

1. Set the following registry entry on each Controller in the Site:

HKEY_LOCAL_MACHINE\Software\Citrix\DesktopServer

Name: AllowMultipleRemotePCAssignments

Type: REG_DWORD

Data: 0 = Disable multiple user assignment, 1 = (Default) Enable multiple user assignment.

2. If there are any existing user assignments, remove them using SDK commands for the VDA to subsequently be eligible for a single automatic assignment.
 1. Remove all assigned users from the VDA: `$machine.AssociatedUserNames | %{ Remove-BrokerUser-Name $_ -Machine $machine }`
 2. Remove the VDA from the Delivery Group: `$machine | Remove-BrokerMachine -DesktopGroup $desktopGroup`
3. Restart the physical office PC.

委派管理

May 04, 2016

“委派管理”模型可以使用角色和基于对象的控制机制灵活地与组织所需的管理活动委派方式相匹配。委派管理可以适应所有规模的部署，并且随着部署复杂性的增加，可以帮助您更细化地配置权限。委派管理基于三个概念：管理员、角色和作用域。

- **管理员** — 管理员是指由 Active Directory 帐户所标识的一个或一组人。每个管理员均与一个或多个角色和作用域对相关联。
- **角色** — 角色代表一项工作职能，并且具有定义的关联权限。例如，“交付组管理员”角色具有“创建交付组”和“从交付组中删除桌面”等权限。管理员在一个站点中可以具有多个角色，因此一个人既可以是交付组管理员，又可以是计算机目录管理员。角色可以内置或自定义。

内置角色包括：

角色	权限
完全权限管理员	可以执行全部任务和操作。完全权限管理员始终与“全部”作用域结合。
只读权限管理员	可以查看指定作用域内的所有对象以及全局信息，但无法更改任何内容。例如，作用域为“伦敦”的只读权限管理员可以查看所有全局对象（例如配置日志记录）和所有伦敦作用域内的对象（例如，伦敦交付组）。但是，该管理员无法查看“纽约”作用域（假设“纽约”作用域和“伦敦”作用域无重叠）中的对象。
技术支持管理员	可以查看交付组，并可以管理与这些组关联的会话和计算机。可以查看所监控的交付组的计算机目录和主机信息，还可以对这些交付组中的计算机执行会话管理和计算机电源管理操作。
计算机目录管理员	可以创建并管理计算机目录，并将计算机置备到这些目录中。可以从虚拟化基础结构、Provisioning Services 和物理机构建计算机目录。此角色可以管理基础映像并安装软件，但不可以向用户分配应用程序或桌面。
交付组管理员	可以交付应用程序、桌面和计算机；还可以管理关联的会话，以及应用程序和桌面配置，如策略和电源管理设置。
主机管理员	可以管理主机连接及其关联的资源设置。可以向用户交付计算机、应用程序或桌面。

在某些产品版本中，您可以根据组织的要求创建自定义角色，并可以更细致地委派权限。您可以使用自定义角色按照控制台中操作或任务的粒度来分配权限。

- **作用域** — 一个作用域代表一个对象集合。作用域用来根据组织的具体情况将对象分组（例如，销售团队使用的交付组集合）。对象可以属于多个作用域；可以将对象视为带有多个作用域的标记。系统提供一个内置的作用域“全部”，其包含全部对象。“完全权限管理员”角色始终与“全部”作用域配对。

XYZ 公司决定根据部门（会计、销售和仓库）管理应用程序和桌面以及桌面操作系统（Windows 7 或 Windows 8）。管理员创建了五个作用域，并分别为每个交付组标记了两个作用域：一个用于所用的部门，另一个用于所用的操作系统。

创建了以下管理员：

管理员	角色	作用域
domain/fred	完全权限管理员	全部（完全权限管理员角色始终拥有全部作用域）
domain/rob	只读权限管理员	全部
domain/heidi	只读权限管理员 技术支持管理员	全部 销售
domain/warehouseadmin	技术支持管理员	仓库
domain/peter	交付组管理员 计算机目录管理员	Win7

- Fred 是完全权限管理员，可以查看、编辑和删除系统中的所有对象。
- Rob 可以查看站点中的所有对象，但无法编辑或删除对象。
- Heidi 可以查看所有对象并可以对销售作用域中的交付组执行技术支持任务。因此她可以管理与这些组关联的会话和计算机，但无法对交付组执行更改，如添加或删除计算

机。

- WarehouseAdmins Active Directory 安全组成员中的任何人都可以查看“仓库”作用域中的计算机，并对这些计算机执行技术支持任务。
- Peter 是 Windows 7 专员，可以管理所有 Windows 7 计算机目录，还可以交付 Windows 7 应用程序、桌面和计算机，无论它们属于哪个部门作用域。管理员曾考虑让 Peter 成为 Win7 作用域的完全权限管理员；但是，她决定不这样做，因为完全权限管理员对不属于作用域范围的所有对象（例如“站点”和“管理员”）也具有完全权限。

一般而言，管理员的数量及其权限的粒度取决于部署的规模和复杂性。

- 对于小规模部署或概念验证部署，一个或几个管理员便足以完成一切工作，无需委派。在这种情况下，将每个管理员均创建为具有内置“完全权限管理员”角色，该角色拥有“全部”作用域。
- 在包含更多计算机、应用程序和桌面的较大规模部署中，需要更多委派。多个管理员的职责（角色）划分可能更为明确。例如，设置两个完全权限管理员，其他则是技术支持管理员。另外，管理员可能只管理特定的对象组（作用域），如计算机目录。在这种情况下，创建新的作用域，并创建具有内置角色之一及适当作用域的管理员。
- 更大规模的部署可能需要更多（或更明确）的作用域，以及具有非常规角色的不同管理员。在这种情况下，编辑或创建更多作用域，创建自定义角色，并根据内置或自定义角色创建每个管理员以及现有和新的作用域。

为实现配置灵活性和方便性，可以在创建管理员时创建新的作用域。另外，还可以在创建或编辑计算机目录或连接时指定作用域。

以本地管理员身份创建站点时，您的用户帐户将自动成为对所有对象具有完全权限的“完全权限管理员”。站点创建完成之后，本地管理员不具有任何特殊权限。

完全权限管理员角色始终具有“全部”范围，此范围无法更改。

默认情况下启用管理员。如果现在要创建新管理员，但此人要在之后某个时间才开始履行管理员职责，则禁用管理员可能很有必要。对于已启用的现有管理员，重新组织对象/范围时，您可能需要禁用多个管理员，当准备启用更新配置时，再将其重新启用。如果禁用管理员会导致没有启用的完全权限管理员，将不能禁用此管理员。在创建、复制或编辑管理员时，启用/禁用复选框将处于可用状态。

在复制、编辑或删除管理员时，如果删除角色/作用域对，此操作将仅删除此管理员的角色与作用域之间的关系，而不会删除角色或作用域本身，也不会影响配置了此角色/作用域对其他任何管理员。

要管理管理员，请在 Studio 导航窗格中单击配置 > 管理员，然后单击中上方窗格中的管理员选项卡。

- 要创建管理员，请在“操作”窗格中单击 Create new Administrator（创建新管理员）。键入或浏览到用户帐户名称，选择或创建一个作用域并选择一个角色。默认情况下，新管理员处于启用状态；可以对此进行更改。
- 要复制管理员，请在中间窗格中选择管理员，然后在“操作”窗格中单击复制管理员。键入或浏览到用户帐户名称。可以选择任何角色/作用域对，然后进行编辑或删除，也可以添加新的角色/作用域对。默认情况下，新管理员处于启用状态；可以对此进行更改。
- 要编辑管理员，请在中间窗格中选择管理员，然后在“操作”窗格中单击编辑管理员。可以编辑或删除任何角色/作用域对，也可以添加新的角色/作用域对。
- 要删除管理员，请在中间窗格中选择管理员，然后在“操作”窗格中单击删除管理员。如果删除管理员会导致没有启用的完全权限管理员，将不能删除此管理员。

角色名称最多可以包含 64 个 Unicode 字符，并且不能包含以下字符：\（反斜线）、/（正斜线）、;（分号）、:（冒号）、#（英镑符号）、,（逗号）、*（星号）、?（问号）、=（等号）、<（左箭头）、>（右箭头）、|（竖线）、[]（左右方括号）、()（左右圆括号）、"（引号）和'（单引号）。说明最多可以包含 256 个 Unicode 字符。

无法编辑或删除内置角色。无法删除任意管理员正在使用的自定义角色。

注意：只有特定产品版本支持自定义角色。不支持自定义角色的版本在“操作”窗格中没有相关项。

要管理角色，请在 Studio 导航窗格中单击配置 > 管理员，然后单击中上方窗格中的角色选项卡。

- 要查看角色详细信息，请在中间窗格中选择角色。中间窗格下方列出了对象类型和角色的相关权限。在下方窗格中单击管理员选项卡，以显示当前具有此角色的管理员列表。
- 要创建自定义角色，请在“操作”窗格中单击创建新角色。输入名称和说明。选择对象类型和权限。
- 要复制角色，请在中间窗格中选择角色，然后在“操作”窗格中单击复制角色。根据需要更改名称、说明、对象类型和权限。
- 要编辑自定义角色，请在中间窗格中选择角色，然后在“操作”窗格中单击编辑角色。根据需要更改名称、说明、对象类型和权限。
- 要删除自定义角色，请在中间窗格中选择角色，然后在“操作”窗格中单击删除角色。出现提示时，确认删除。

创建站点时，唯一可用的作用域是不能删除的“全部”作用域。

可以使用以下过程创建作用域。也可以在创建管理员时创建作用域；每个管理员必须至少与一个角色和作用域对相关。创建或编辑桌面、计算机目录、应用程序或主机时，可将其添加到现有作用域；如果未将其添加到作用域，则它们仍是“全部”作用域的一部分。

站点创建和委派管理员对象（作用域和角色）均无法归入作用域内。但是，无法归入作用域内的对象可包含在“全部”作用域内。（具有完全权限的管理员始终具有“全部”作用域。）计算机、电源操作、桌面和会话不会直接归入作用域；管理员可通过相关的计算机目录或交付组分配对这些对象的权限。

作用域名称最多可以包含 64 个 Unicode 字符，并且不能包含以下字符：\（反斜线）、/（正斜线）、;（分号）、:（冒号）、#（英镑符号）、,（逗号）、*（星号）、?（问号）、=（等号）、<（左箭头）、>（右箭头）、|（竖线）、[]（左右方括号）、()（左右圆括号）、"（引号）和'（单引号）。说明最多可以包含 256 个 Unicode 字符。

在复制或编辑作用域时，请记住，从作用域中删除对象会导致管理员无法访问这些对象。如果编辑的作用域与一个或多个角色配对，请确保对作用域所做的更新不会使任何角色/作用域对无法使用。

要管理作用域，请在 Studio 导航窗格中单击配置 > 管理员，然后单击中上方窗格中的作用域选项卡。

- 要创建作用域，请在“操作”窗格中单击创建新作用域。输入名称和说明。要包括特定类型的所有对象（如交付组），请选择对象类型。要包括特定对象，请展开类型，然后选择各个对象（例如，销售团队使用的各个交付组）。
- 要复制作用域，请在中间窗格中选择作用域，然后在“操作”窗格中单击复制作用域。输入名称和说明。根据需要更改对象类型和对象。
- 要编辑作用域，请在中间窗格中选择作用域，然后在“操作”窗格中单击编辑作用域。根据需要更改名称、说明、对象类型和对象。
- 要删除作用域，请在中间窗格中选择作用域，然后在“操作”窗格中单击删除作用域。出现提示时，确认删除。

可以创建两种类型的委派管理报告：

- HTML 报告，此报告将列出与管理员关联的角色/作用域对以及每种对象类型（例如，交付组和计算机目录）的各个权限。通过 Studio 生成此报告。要创建此报告，请在导航窗格中单击配置 > 管理员。在中间窗格中选择管理员，然后在“操作”窗格中单击创建报告。

您还可以在创建、复制或编辑管理员时请求此报告。

- 将所有内置和自定义角色映射到权限的 HTML 或 CSV 报告。通过运行名为 OutputPermissionMapping.ps1 的 PowerShell 脚本生成此报告。要运行此脚本，您必须是完全权限管理员、只读权限管理员或具有读取角色权限的自定义管理员。此脚本位于：Program Files\Citrix\DelegatedAdmin\SnapIn\Citrix.DelegatedAdmin.Admin.V1\Scripts\。

语法：

```
OutputPermissionMapping.ps1 [-Help] [-Csv] [-Path <string>] [-AdminAddress <string>] [-Show] [<通用参数>]
```

参数	说明
-Help	显示脚本帮助。
-Csv	指定 CSV 输出。默认值：HTML
-Path <字符串>	输出的写入位置。默认值：stdout
-AdminAddress <字符串>	要连接的 Delivery Controller 的 IP 地址或主机名。默认名称为 localhost
-Show	（仅当指定了 -Path 参数时此参数才有效）将输出写入到文件时，-Show 会在相应的程序中打开此输出，例如 Web 浏览器。
<通用参数>	Verbose、Debug、ErrorAction、ErrorVariable、WarningAction、WarningVariable、OutBuffer 和 OutVariable。有关详细信息，请参阅 Microsoft 文档。

以下示例将 HTML 表写入到名为 Roles.html 的文件，并在 Web 浏览器中打开此表。

```
& "$env:ProgramFiles\Citrix\DelegatedAdmin\SnapIn\Citrix.DelegatedAdmin.Admin.V1\Scripts\OutputPermissionMapping.ps1" -Path Roles.html -Show
```

以下示例将 CSV 表写入到名为 Roles.csv 的文件。未显示此表。

```
& "$env:ProgramFiles\Citrix\DelegatedAdmin\SnapIn\Citrix.DelegatedAdmin.Admin.V1\Scripts\OutputPermissionMapping.ps1" -CSV -Path Roles.csv
```

在 Windows 命令提示窗口中，上例命令为：

```
powershell -command "& '%ProgramFiles%\Citrix\DelegatedAdmin\SnapIn\Citrix.DelegatedAdmin.Admin.V1\Scripts\OutputPermissionMapping.ps1' -CSV -Path Roles.csv"
```

智能卡

Nov 02, 2016

Smart cards and equivalent technologies are supported within the guidelines described in this article. To use smart cards with XenApp or XenDesktop:

- Understand your organization's security policy concerning the use of smart cards. These policies might, for example, state how smart cards are issued and how users should safeguard them. Some aspects of these policies might need to be reassessed in a XenApp or XenDesktop environment.
- Determine which user device types, operating systems, and published applications are to be used with smart cards.
- Familiarize yourself with smart card technology and your selected smart card vendor hardware and software.
- Know how to deploy digital certificates in a distributed environment.

Types of smart cards

Enterprise and consumer smart cards have the same dimensions, electrical connectors, and fit the same smart card readers.

Smart cards for enterprise use contain digital certificates. These smart cards support Windows logon, and can also be used with applications for digital signing and encryption of documents and e-mail. XenApp and XenDesktop support these uses.

Smart cards for consumer use do not contain digital certificates; they contain a shared secret. These smart cards can support payments (such as a chip-and-signature or chip-and-PIN credit card). They do not support Windows logon or typical Windows applications. Specialized Windows applications and a suitable software infrastructure (including, for example, a connection to a payment card network) are needed for use with these smart cards. Contact your Citrix representative for information on supporting these specialized applications on XenApp or XenDesktop.

For enterprise smart cards, there are compatible equivalents that can be used in a similar way.

- A smart card-equivalent USB token connects directly to a USB port. These USB tokens are usually the size of a USB flash drive, but can be as small as a SIM card used in a mobile phone. They appear as the combination of a smart card plus a USB smart card reader.
- A virtual smart card using a Windows Trusted Platform Module (TPM) appears as a smart card. These virtual smart cards are supported for Windows 8 and Windows 10, using Citrix Receiver minimum 4.3.
 - Versions of XenApp and XenDesktop earlier than 7.6 FP3 do not support virtual smart cards.
 - For more information on virtual smart cards, see [Virtual Smart Card Overview](#).

Note: The term "virtual smart card" is also used to describe a digital certificate simply stored on the user computer. These digital certificates are not strictly equivalent to smart cards.

XenApp and XenDesktop smart card support is based on the Microsoft Personal Computer/Smart Card (PC/SC) standard specifications. A minimum requirement is that smart cards and smart card devices must be supported by the underlying Windows operating system and must be approved by the Microsoft Windows Hardware Quality Labs (WHQL) to be used on computers running qualifying Windows operating systems. See the Microsoft documentation for additional information about hardware PC/SC compliance. Other types of user devices may comply with the PS/SC standard. For more information, refer to the Citrix Ready program at <http://www.citrix.com/ready/>.

Usually, a separate device driver is needed for each vendor's smart card or equivalent. However, if smart cards conform to a

standard such as the NIST Personal Identity Verification (PIV) standard, it may be possible to use a single device driver for a range of smart cards. The device driver must be installed on both the user device and the Virtual Delivery Agent (VDA). The device driver is often supplied as part of a smart card middleware package available from a Citrix partner; the smart card middleware package will offer advanced features. The device driver may also be described as a Cryptographic Service Provider (CSP), Key Storage Provider (KSP), or minidriver.

The following smart card and middleware combinations for Windows systems have been tested by Citrix as representative examples of their type. However, other smart cards and middleware can also be used. For more information about Citrix-compatible smart cards and middleware, see <http://www.citrix.com/ready>.

Middleware	Matching cards
ActivClient 7.0 (DoD mode enabled)	DoD CAC card
ActivClient 7.0 in PIV mode	NIST PIV card
Microsoft mini driver	NIST PIV card
GemAlto Mini Driver for .NET card	GemAlto .NET v2+
Microsoft native driver	Virtual Smart Cards (TPM)

For information about smart card usage with other types of devices, see the Citrix Receiver documentation for that device. For more information about PIV usage with XenDesktop, see [Configuring Citrix XenDesktop 7.6 and NetScaler Gateway 10.5 with PIV Smart Card Authentication](#).

For information about smart card usage with other types of devices, see the Citrix Receiver documentation for that device.

Smart cards are supported only for remote access to physical office PCs running Windows 10, Windows 8 or Windows 7; smart cards are not supported for office PCs running Windows XP.

The following smart cards were tested with Remote PC Access:

Middleware	Matching cards
Gemalto .NET minidriver	Gemalto .NET v2+
ActivIdentity ActivClient 6.2	NIST PIV
ActivIdentity ActivClient 6.2	CAC
Microsoft minidriver	NIST PIV

Microsoft native driver	Virtual smart cards

Types of smart card readers

A smart card reader may be built in to the user device, or be separately attached to the user device (usually via USB or Bluetooth). Contact card readers that comply with the USB Chip/Smart Card Interface Devices (CCID) specification are supported. They contain a slot or swipe into which the user inserts the smart card. The Deutsche Kreditwirtschaft (DK) standard defines four classes of contact card readers.

- Class 1 smart card readers are the most common, and usually just contain a slot. Class 1 smart card readers are supported, usually with a standard CCID device driver supplied with the operating system.
- Class 2 smart card readers also contain a secure keypad that cannot be accessed by the user device. Class 2 smart card readers may be built into a keyboard with an integrated secure keypad. For class 2 smart card readers, contact your Citrix representative; a reader-specific device driver may be required to enable the secure keypad capability.
- Class 3 smart card readers also contain a secure display. Class 3 smart card readers are not supported.
- Class 4 smart card readers also contain a secure transaction module. Class 4 smart card readers are not supported.

Note: The smart card reader class is unrelated to the USB device class.

Smart card readers must be installed with a corresponding device driver on the user device.

User experience

Smart card support is integrated into XenApp and XenDesktop, using a specific ICA/HDX smart card virtual channel that is enabled by default.

Important: Do not use generic USB redirection for smart card readers. This is disabled by default for smart card readers, and is not supported if enabled.

Multiple smart cards and multiple readers can be used on the same user device, but if pass-through authentication is in use, only one smart card must be inserted when the user starts a virtual desktop or application. When a smart card is used within an application (for example, for digital signing or encryption functions), there might be additional prompts to insert a smart card or enter a PIN. This can occur if more than one smart card has been inserted at the same time.

- If users are prompted to insert a smart card when the smart card is already in the reader, they should select Cancel.
- If users are prompted for the PIN, they should enter the PIN again.

If you are using hosted applications running on Windows Server 2008 or 2008 R2 and with smart cards requiring the Microsoft Base Smart Card Cryptographic Service Provider, you might find that if a user runs a smart card transaction, all other users who use a smart card in the logon process are blocked. For further details and a hotfix for this issue, see <http://support.microsoft.com/kb/949538>.

You can reset PINs using a card management system or vendor utility.

Before deploying smart cards

- Obtain a device driver for the smart card reader and install it on the user device. Many smart card readers can use the CCID device driver supplied by Microsoft.
- Obtain a device driver and cryptographic service provider (CSP) software from your smart card vendor, and install them on both user devices and virtual desktops. The driver and CSP software must be compatible with XenApp and XenDesktop; check the vendor documentation for compatibility. For virtual desktops using smart cards that support and use the minidriver model, smart card minidrivers should download automatically, but you can obtain them from <http://catalog.update.microsoft.com> or from your vendor. Additionally, if PKCS#11 middleware is required, obtain it from the card vendor.
- **Important:** Citrix recommends that you install and test the drivers and CSP software on a physical computer before installing Citrix software.
- Add the Citrix Receiver for Web URL to the Trusted Sites list for users who work with smart cards in Internet Explorer with Windows 10. In Windows 10, Internet Explorer does not run in protected mode by default for trusted sites.
- Ensure that your public key infrastructure (PKI) is configured appropriately. This includes ensuring that certificate-to-account mapping is correctly configured for Active Directory environment and that user certificate validation can be performed successfully.
- Ensure your deployment meets the system requirements of the other Citrix components used with smart cards, including Citrix Receiver and StoreFront.
- Ensure access to the following servers in your Site:
 - The Active Directory domain controller for the user account that is associated with a logon certificate on the smart card
 - Delivery Controller
 - Citrix StoreFront
 - Citrix NetScaler Gateway/Citrix Access Gateway 10.x
 - VDA
 - (Optional for Remote PC Access): Microsoft Exchange Server

Enable smart card use

Step 1. Issue smart cards to users according to your card issuance policy.

Step 2. (Optional) Set up the smart cards to enable users for Remote PC Access.

Step 3. Install and configure the Delivery Controller and StoreFront (if not already installed) for smart card remoting.

Step 4. Enable StoreFront for smart card use. For details, see [Configure smart card authentication in the StoreFront documentation](#).

Step 5. Enable NetScaler Gateway/Access Gateway for smart card use. For details, see [Configuring Authentication and Authorization and Configuring Smart Card Access with the Web Interface in the NetScaler documentation](#).

Step 6. Enable VDAs for smart card use.

- Ensure the VDA has the required applications and updates.
- Install the middleware.

- Set up smart card remoting, enabling the communication of smart card data between Citrix Receiver on a user device and a virtual desktop session.

Step 7. Enable user devices (including domain-joined or non-domain-joined machines) for smart card use. See Configure smart card authentication in the StoreFront documentation for details.

- Import the certificate authority root certificate and the issuing certificate authority certificate into the device's keystore.
- Install your vendor's smart card middleware.
- Install and configure Citrix Receiver for Windows, being sure to import `icaclient.adm` using the Group Policy Management Console and enable smart card authentication.

Step 8. Test the deployment. Ensure that the deployment is configured correctly by launching a virtual desktop with a test user's smart card. Test all possible access mechanisms (for example, accessing the desktop through Internet Explorer and Citrix Receiver).

智能卡部署

May 04, 2016

本产品版本和包含本版本的混合环境支持下列类型的智能卡部署。其他配置可能可以运行，但不受支持。

类型	StoreFront 连接
加入域的本地计算机	直接连接
从加入域的计算机远程访问	通过 NetScaler Gateway 连接
未加入域的计算机	直接连接
从未加入域的计算机远程访问	通过 NetScaler Gateway 连接
访问桌面设备站点的未加入域的计算机和瘦客户端	通过桌面设备站点连接
通过 XenApp Services URL 访问 StoreFront 的加入域的计算机和瘦客户端	通过 XenApp Services URL 连接

部署类型由智能卡读卡器连接到的用户设备的以下特性定义：

- 设备是否已加入域。
- 设备连接到 StoreFront 的方式。
- 查看虚拟桌面和应用程序使用的软件。

此外，启用智能卡的应用程序（如 Microsoft Word 和 Microsoft Excel）也可在这些部署中使用。这些应用程序允许用户对文档进行数字签名或加密。

在其中的每个部署中，如有可能，Receiver 支持双模式身份验证，允许用户在使用智能卡和输入其用户名和密码之间进行选择。如果无法使用智能卡（例如，用户将智能卡遗忘在家中或登录证书已过期），此功能会很有帮助。

由于未加入域的设备的用户将直接登录到 Receiver for Windows，因此，您可以允许用户回退至显式身份验证。如果您配置了双模式身份验证，则系统最初会提示用户使用智能卡和 PIN 进行登录，但在智能卡出现问题时可以选择使用显式身份验证。

如果您部署 NetScaler Gateway，则用户登录到设备后，Receiver for Windows 会提示用户完成 NetScaler Gateway 的身份验证。对于加入域的设备 and 未加入域的设备均是如此。用户可以使用智能卡和 PIN 或使用显式凭据登录到 NetScaler Gateway。这样，您可以向用户提供 NetScaler Gateway 登录的双模式身份验证。可以配置从 NetScaler Gateway 到 StoreFront 的直通身份验证，并将智能卡用户的凭据验证委派给 NetScaler Gateway，这样用户就可以无提示地通过 StoreFront 的身份验证。

在 Citrix 环境中，在单个林中支持智能卡。跨林进行智能卡登录要求对所有用户帐户启用直接双向林信任。不支持涉及智能卡的更加复杂的多林部署（即，其中的信任仅为单向信任或具有不同的类型）。

您可以在包括远程桌面的 Citrix 环境中使用智能卡。可以在本地（在智能卡连接的用户设备上）或远程（在用户设备连接的远

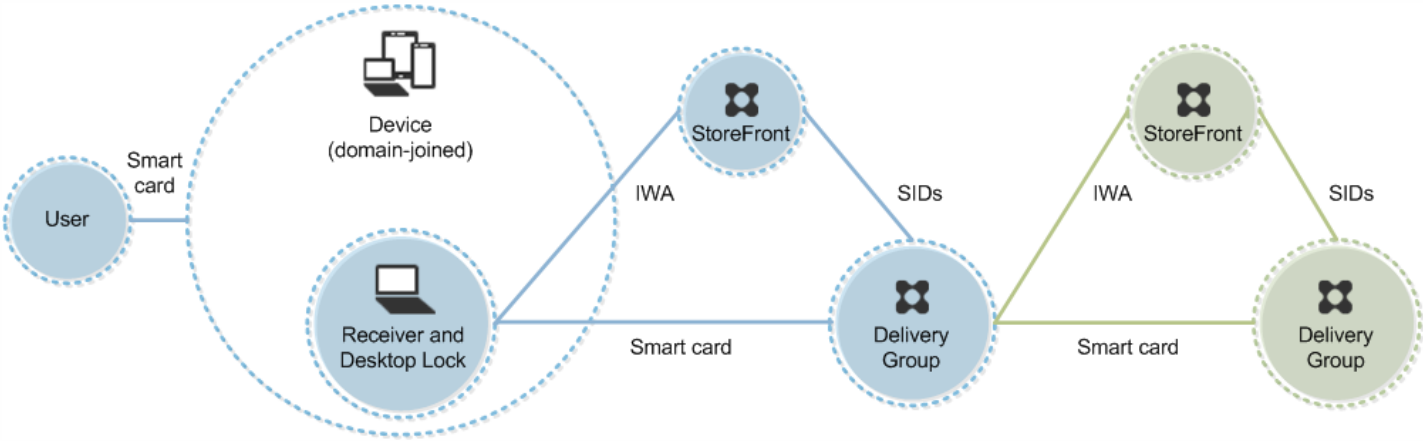
程桌面上) 安装此功能。

在产品上设置的智能卡移除策略用于确定当在会话期间从读卡器中移除智能卡时所发生的操作。智能卡移除策略通过 Windows 操作系统进行配置，并且也由 Windows 操作系统来处理。

策略设置	桌面行为
不采取任何措施	不采取任何措施。
锁定工作站	桌面会话断开连接，并锁定虚拟桌面。
强制注销	将强制用户注销。如果网络连接已断开，并启用了此设置，则此会话可能会注销，用户可能会丢失数据。
如果是远程终端服务会话，则断开连接	会话断开连接，并锁定虚拟桌面。

如果启用证书吊销检查，并且用户将具有无效证书的智能卡插入读卡器，用户将无法对与该证书相关的桌面或应用程序进行身份验证或访问。例如，如果使用无效证书进行电子邮件解密，电子邮件将保持加密状态。如果卡上的其他证书（例如，用于身份验证的证书）仍有效，这些功能将仍有效。

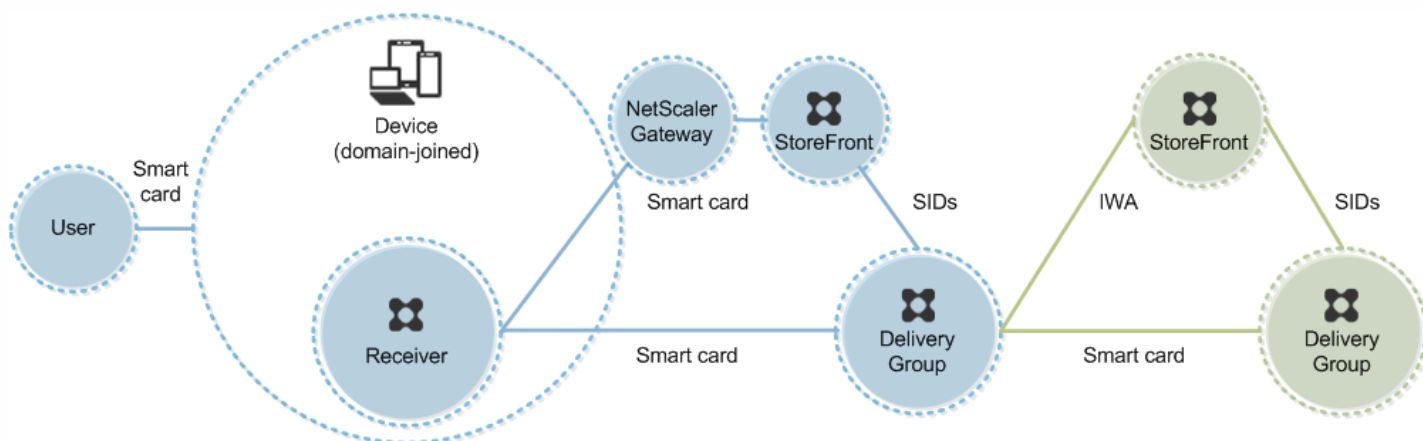
此部署涉及运行 Desktop Viewer 并直接连接到 StoreFront 的已加入域的用户设备。



用户使用智能卡和 PIN 登录设备。Receiver 使用集成 Windows 身份验证 (IWA) 向 StoreFront 服务器进行用户身份验证。StoreFront 将用户安全标识符 (SID) 传递到 XenApp 或 XenDesktop。当用户启动虚拟桌面或应用程序时，系统不会提示用户重新输入 PIN，因为 Receiver 上已配置单点登录功能。

通过添加第二台 StoreFront 服务器和托管应用程序的服务器，此部署可以扩展到双跳。虚拟桌面的 Receiver 对第二台 StoreFront 服务器进行身份验证。第二次连接时，可以使用任何身份验证方法。第一个跃点显示的配置可在第二个跃点中重新使用，或仅能在第二个跃点中使用。

此部署涉及运行 Desktop Viewer 并通过 NetScaler Gateway/Access Gateway 连接到 StoreFront 的已加入域的用户设备。

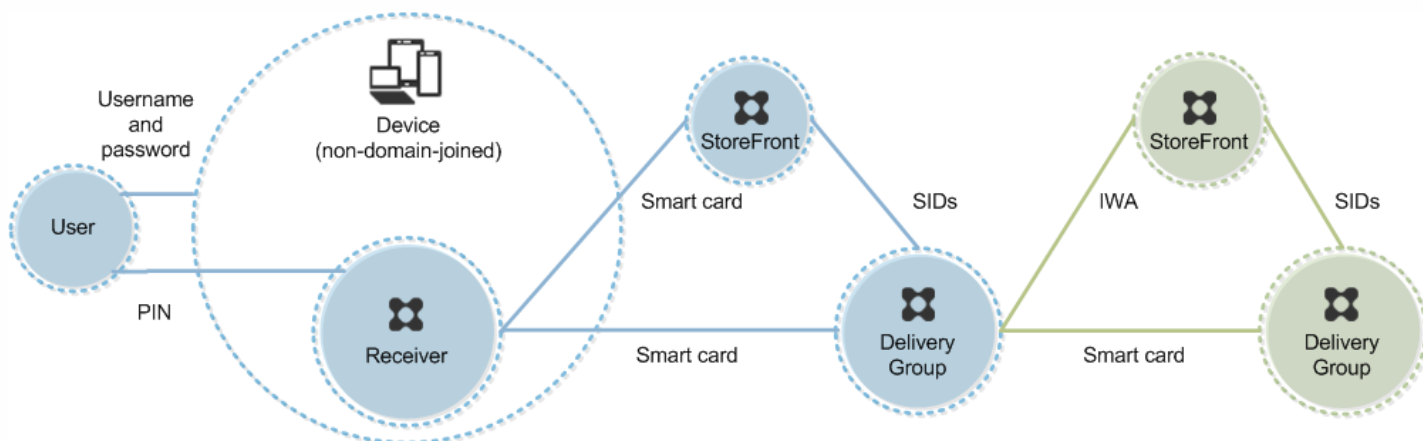


用户使用智能卡和 PIN 登录设备，然后重新登录到 NetScaler Gateway/Access Gateway。第二次登录可以使用智能卡和 PIN，也可以使用用户名和密码，因为在此部署中 Receiver 允许双模式身份验证。

用户自动登录 StoreFront，将用户安全标识符 (SID) 传递到 XenApp 或 XenDesktop。当用户启动虚拟桌面或应用程序时，系统不会提示用户重新输入 PIN，因为 Receiver 上已配置单点登录功能。

通过添加第二台 StoreFront 服务器和托管应用程序的服务器，此部署可以扩展到双跳。虚拟桌面的 Receiver 对第二台 StoreFront 服务器进行身份验证。第二次连接时，可以使用任何身份验证方法。第一个跃点显示的配置可在第二个跃点中重新使用，或仅能在第二个跃点中使用。

此部署涉及运行 Desktop Viewer 并直接连接到 StoreFront 的未加入域的用户设备。

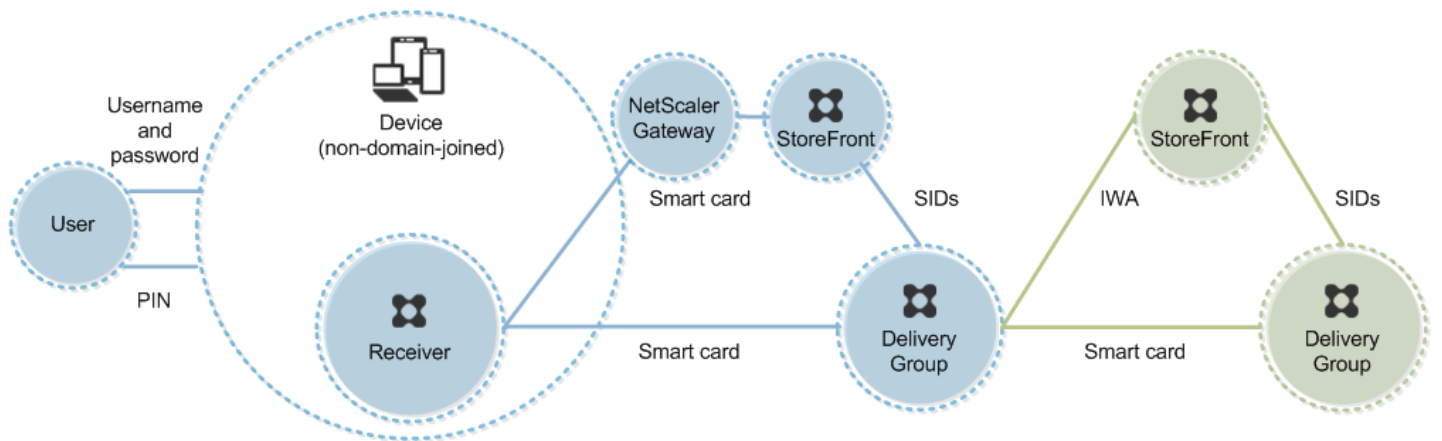


用户登录设备。通常情况下，用户需要输入用户名和密码，但由于此设备未加入域，因此此登录的凭据是可选的。因为在此部署中可使用双模式身份验证，因此 Receiver 会提示用户输入智能卡和 PIN，或使用用户名和密码。然后 Receiver 对 StoreFront 进行身份验证。

StoreFront 将用户安全标识符 (SID) 传递到 XenApp 或 XenDesktop。当用户启动虚拟桌面或应用程序时，系统会提示用户重新输入 PIN，因为在此部署中未提供单点登录功能。

通过添加第二台 StoreFront 服务器和托管应用程序的服务器，此部署可以扩展到双跳。虚拟桌面的 Receiverd 对第二台 StoreFront 服务器进行身份验证。第二次连接时，可以使用任何身份验证方法。第一个跃点显示的配置可在第二个跃点中重新使用，或仅能在第二个跃点中使用。

此部署涉及运行 Desktop Viewer 并直接连接到 StoreFront 的未加入域的用户设备。



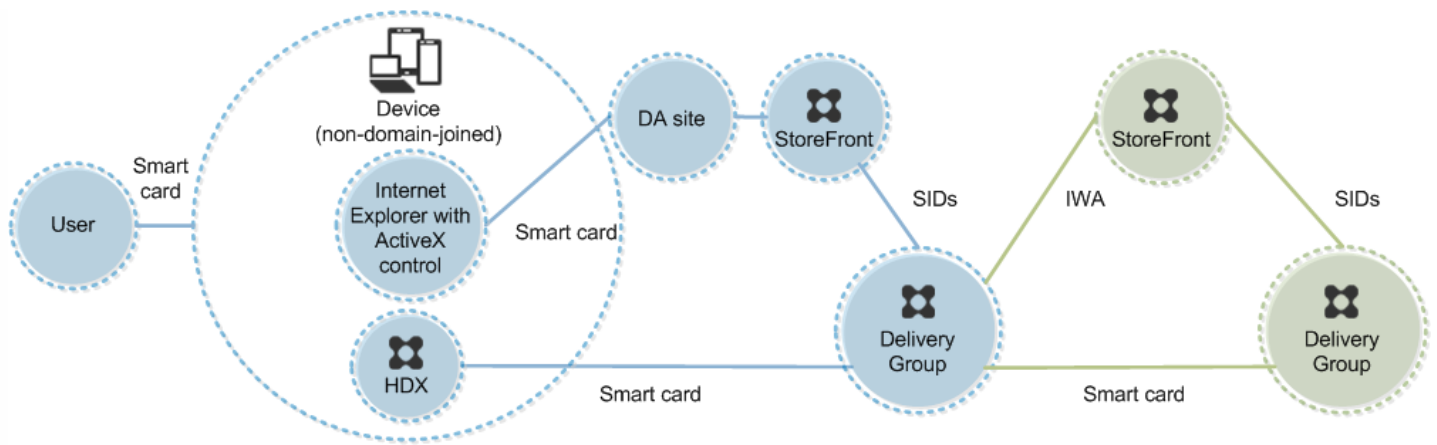
用户登录设备。通常情况下，用户需要输入用户名和密码，但由于此设备未加入域，因此此登录的凭据是可选的。因为在此部署中可使用双模式身份验证，因此 Receiver 会提示用户输入智能卡和 PIN，或使用用户名和密码。然后 Receiver 对 StoreFront 进行身份验证。

StoreFront 将用户安全标识符 (SID) 传递到 XenApp 或 XenDesktop。当用户启动虚拟桌面或应用程序时，系统会提示用户重新输入 PIN，因为在此部署中未提供单点登录功能。

通过添加第二台 StoreFront 服务器和托管应用程序的服务器，此部署可以扩展到双跳。虚拟桌面的 Receiverd 对第二台 StoreFront 服务器进行身份验证。第二次连接时，可以使用任何身份验证方法。第一个跃点显示的配置可在第二个跃点中重新使用，或仅能在第二个跃点中使用。

此部署涉及运行 Desktop Lock，并通过桌面设备站点连接到 StoreFront 的未加入域的用户设备。

Desktop Lock 是随 XenApp、XenDesktop 和 VDI-in-a-Box 一起发布的独立组件。它是 Desktop Viewer 的替代项，主要是针对重新设计用途的 Windows 计算机和 Windows 瘦客户端而设计的。Desktop Lock 取代了这些用户设备中的 Windows shell 和任务管理器，以阻止用户访问基础设备。通过使用 Desktop Lock，用户可以访问 Windows Server 计算机桌面和 Windows 桌面计算机桌面。可以选择安装 Desktop Lock。

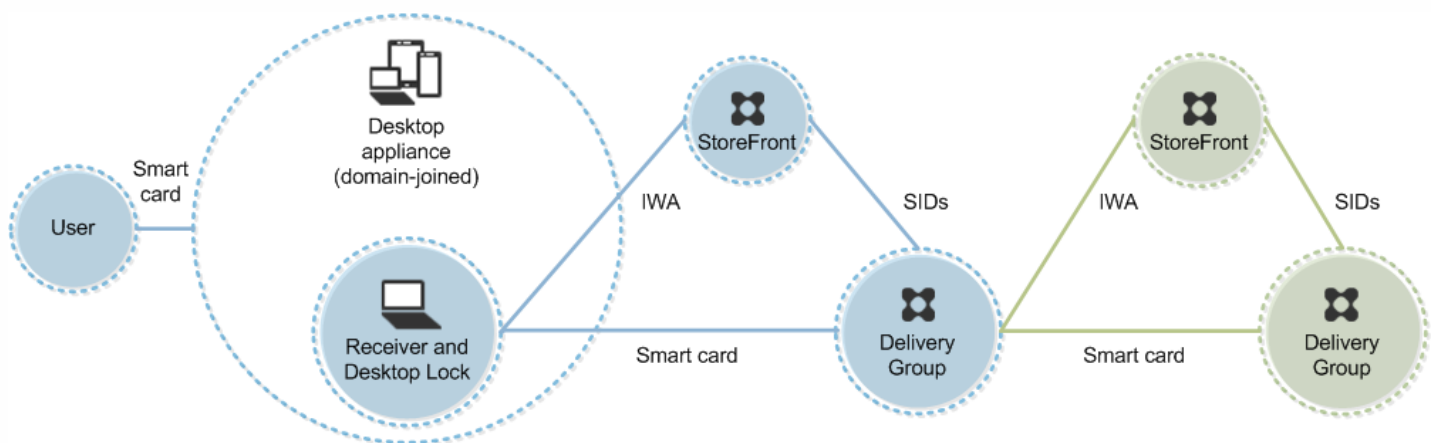


用户使用智能卡登录设备。如果 Desktop Lock 正在设备上运行，该设备配置为通过在 Kiosk 模式下运行的 Internet Explorer 启动桌面设备站点。该站点上的 ActiveX 控件会提示用户输入 PIN，然后将其发送到 StoreFront。StoreFront 将用户安全标识符 (SID) 传递到 XenApp 或 XenDesktop。分配的桌面组列表（按字母顺序）中的第一个可用桌面将启动。

通过添加第二台 StoreFront 服务器和托管应用程序的服务器，此部署可以扩展到双跳。虚拟桌面的 Receiver 对第二台 StoreFront 服务器进行身份验证。第二次连接时，可以使用任何身份验证方法。第一个跃点显示的配置可在第二个跃点中重新使用，或仅能在第二个跃点中使用。

此部署涉及运行 Desktop Lock，并通过 XenApp Services URL 连接到 StoreFront 的加入域的用户设备。

Desktop Lock 是随 XenApp、XenDesktop 和 VDI-in-a-Box 一起发布的独立组件。它是 Desktop Viewer 的替代项，主要是针对重新设计用途的 Windows 计算机和 Windows 瘦客户端而设计的。Desktop Lock 取代了这些用户设备中的 Windows shell 和任务管理器，以阻止用户访问基础设备。通过使用 Desktop Lock，用户可以访问 Windows Server 计算机桌面和 Windows 桌面计算机桌面。可以选择安装 Desktop Lock。



用户使用智能卡和 PIN 登录设备。如果 Desktop Lock 正在设备上运行，它会使用集成 Windows 身份验证 (IWA) 向 StoreFront 服务器进行用户身份验证。StoreFront 将用户安全标识符 (SID) 传递到 XenApp 或 XenDesktop。当用户启动虚拟桌面时，系统不会提示用户重新输入 PIN，因为 Receiver 上已配置单点登录功能。

通过添加第二台 StoreFront 服务器和托管应用程序的服务器，此部署可以扩展到双跳。虚拟桌面的 Receiver 对第二台 StoreFront 服务器进行身份验证。第二次连接时，可以使用任何身份验证方法。第一个跃点显示的配置可在第二个跃点中重新使用，或仅能在第二个跃点中使用。

使用智能卡进行直通身份验证和单点登录

May 04, 2016

运行 Windows 10、Windows 8 和 Windows 7 SP1 Enterprise Edition 和 Professional Edition 的用户设备支持使用智能卡对虚拟桌面进行直通身份验证。

运行 Windows Server 2008 和 Windows Server 2012 的服务器支持使用智能卡对托管应用程序进行直通身份验证。

要使用智能卡对托管应用程序进行直通身份验证，请确保在配置使用智能卡进行直通身份验证作为站点的身份验证方法时启用 Kerberos。

注意：使用智能卡进行直通身份验证的可用性取决于许多因素，包括但不限于以下因素：

- 贵组织关于直通身份验证的安全策略。
- 中间件类型和配置。
- 智能卡读卡器类型。
- 中间件 PIN 缓存策略。

Citrix StoreFront 上已配置使用智能卡进行直通身份验证。请参阅 StoreFront 文档中的

— 配置身份验证服务

以了解详细信息。

单点登录是一项 Citrix 功能，用于实现对虚拟桌面和应用程序启动的直通身份验证。您可以在加入域的直接访问 StoreFront 以及加入域的通过 NetScaler 访问 StoreFront 智能卡部署中使用此功能，以减少用户输入其 PIN 的次数。要在这些部署类型中使用单点登录，请在 default.ica 文件（位于 StoreFront 服务器上）中编辑以下参数：

- 加入域的直接访问 StoreFront 智能卡部署 — 将 DisableCtrlAltDel 设置为 Off
- 加入域的通过 NetScaler 访问 StoreFront 智能卡部署 — 将 UseLocalUserAndPassword 设置为 On

有关设置这些参数的更多说明，请参阅 StoreFront 或 NetScaler Gateway 文档。

单点登录功能的可用性取决于多种因素，包括但不限于以下因素：

- 您的组织关于单点登录的安全策略。
- 中间件类型和配置。
- 智能卡读卡器类型。
- 中间件 PIN 缓存策略。

注意：如果用户在连接智能卡读卡器的计算机上登录到 Virtual Delivery Agent (VDA)，可能会显示一个 Windows 头像，表示上一次成功的身份验证模式，如智能卡或密码。因此，当启用单点登录时，可能会显示单点登录头像。要登录，用户必须选择切换用户以选择另一个头像，因为单点登录头像不起作用。

SSL

Jul 13, 2016

配置 XenApp 或 XenDesktop 站点以使用安全套接字层 (SSL) 安全协议包括以下过程：

- 获取服务器证书并在所有 Delivery Controller 上安装和注册，并使用 SSL 证书配置端口。 有关详细信息，请参阅[在 Controller 上安装 SSL 服务器证书](#)。
(可选) 可以更改 Controller 用于侦听 HTTP 和 HTTPS 流量的端口。
 - 通过完成以下任务在用户和 Virtual Delivery Agents (VDA) 之间启用 SSL 连接：
 - 在安装 VDA 的计算机上配置 SSL。 (方便起见，后面将安装 VDA 的计算机简称为 VDA。) 可以使用 Citrix 提供的 PowerShell 脚本，也可以手动配置。 有关常见信息，请参阅[关于 VDA 上的 SSL 设置](#)。 有关详细信息，请参阅[使用 PowerShell 脚本在 VDA 上配置 SSL](#)和[在 VDA 上手动配置 SSL](#)。
 - 通过在 Studio 中运行一组 PowerShell cmdlet，在包含 VDA 的交付组中配置 SSL。 有关详细信息，请参阅[在交付组上配置 SSL](#)。
- 要求和注意事项：
- 在用户和 VDA 之间启用 SSL 连接仅对 XenApp 7.6 和 XenDesktop 7.6 及后续受支持的版本有效。
 - 在安装组件、创建站点、创建计算机目录和创建交付组之后，在交付组中和 VDA 上配置 SSL。
 - 要在交付组中配置 SSL，必须具有更改 Controller 访问规则的权限。完全权限管理员具有此权限。
 - 要在 VDA 上配置 SSL，必须是安装 VDA 的计算机上的 Windows 管理员。
 - 如果打算在从早期版本升级的 VDA 上配置 SSL，请在升级之前卸载这些计算机上的 SSL Relay 软件。
 - PowerShell 脚本在静态 VDA 上配置 SSL，不会在通过 Machine Creation Services 或 Provisioning Services 置备的池 VDA 上配置 SSL。对于第二种情况，计算机映像在每次重新启动时重置。

对于包括使用 Windows 注册表的任务：

警告：注册表编辑不当会导致严重问题，可能导致需要重新安装操作系统。 Citrix 无法保证因“注册表编辑器”使用不当导致出现的问题能够得以解决。 使用“注册表编辑器”需自担风险。 在编辑注册表之前，请务必进行备份。

有关为站点数据库启用 SSL 的信息，请参阅 [CTX137556](#)。

在 Controller 上安装 SSL 服务器证书

- 对于 HTTPS，XML Service 通过使用服务器证书而非客户端证书来支持 SSL 功能。 要获取证书并在 Controller 上进行安装和注册，以及使用 SSL 证书配置端口，请遵循以下操作：
- 如果 Controller 中安装了 IIS，请按照 <https://technet.microsoft.com/zh-cn/library/cc771438%28v=ws.10%29.aspx> 中的指南操作。
 - 如果控制器中未安装 IIS，配置证书的其中一种方法是：
 1. 按照 <http://blogs.technet.com/b/pki/archive/2009/08/05/how-to-create-a-web-server-ssl-certificate-manually.aspx> 中所述获得 SSL 服务器证书并将其安装在 Controller 上。 有关 certreq 工具的详细信息，请参阅 [http://technet.microsoft.com/en-us/library/cc736326\(W5.10\).aspx](http://technet.microsoft.com/en-us/library/cc736326(W5.10).aspx)。
如果打算使用 PowerShell 脚本在 VDA 上配置 SSL，请确保证书位于证书存储的“本地计算机”>“个人”>“证书”区域，除非您打算指定 SSL 证书的指纹。 如果多个证书位于此位置，将使用找到的第一个证书。
 2. 使用证书配置端口；请参阅 <http://msdn.microsoft.com/zh-cn/library/ms733791%28v=vs.110%29.aspx>。

更改 HTTP 或 HTTPS 端口

默认情况下，Controller 上的 XML Service 在端口 80 上侦听 HTTP 流量，在端口 443 上侦听 HTTPS 流量。 尽管可以使用非默认端口，但请注意：将 Controller 暴露在不受信任的网络上存在安全风险。

部署独立 StoreFront 服务器比更改默认值更可取。

要更改 Controller 使用的默认 HTTP 或 HTTPS 端口，请从 Studio 运行以下命令：BrokerService.exe -WIPORT <http 端口> -WISSLPOR <https 端口>

其中，<http 端口> 是用于 HTTP 流量的端口号，<https 端口> 是用于 HTTPS 流量的端口号。

注意：更改端口后，Studio 可能会显示关于许可证兼容性和升级的消息。 要解决此问题，请使用以下 PowerShell cmdlet 序列重新注册服务实例：

```
Get-ConfigRegisteredServiceInstance -ServiceType Broker -Binding XML_HTTPS | Unregister-ConfigRegisteredServiceInstance
Get-BrokerServiceInstance | where Binding -eq "XML_HTTPS" | Register
```

仅会强制执行 HTTPS 流量

如果希望 XML Service 忽略默认端口上的 HTTP 流量，请在 Controller 上的 HKLM\Software\Citrix\DesktopServer\ 中设置以下注册表值，然后重新启动 Broker Service。

要忽略 HTTP 流量，请将 XmlServicesEnableNonSsl 设置为 0。

提供了一个用于忽略 HTTPS 流量的相应注册表值：XmlServicesEnableSsl。 请确保未将其设置为 0。

关于 VDA 上的 SSL 设置

在 VDA 上配置 SSL 时，会更改已安装 SSL 证书上的权限，向 ICA Service 授予证书私钥的读取权限，并将以下信息告知 ICA Service：

- **证书存储中用于 SSL 的证书。**
- **用于 SSL 连接的 TCP 端口号。**
必须将 Windows 防火墙 (如果启用) 配置为允许此 TCP 端口上的传入连接。 使用 PowerShell 脚本时会完成此配置。
- **允许的 SSL 协议版本。**
支持的 SSL 协议版本遵循以下层次结构 (从最低到最高)：SSL 3.0、TLS 1.0、TLS 1.1 和 TLS 1.2。 指定允许的最低版本；将允许使用此版本或更高版本的所有协议连接。

例如，如果指定 TLS 1.1 作为最低版本，则允许 TLS 1.1 和 TLS 1.2 协议连接。 如果指定 SSL 3.0 作为最低版本，则允许所有受支持版本的连接。 如果指定 TLS 1.2 作为最低版本，则仅允许 TLS 1.2 连接。
- **允许的 SSL 密码。**
密码集是常用 SSL 密码的列表。 客户端连接并发送受支持的 SSL 密码列表时，VDA 将客户端的其中一个密码与其已配置密码集中的其中一个密码相匹配并接受连接。 如果客户端发送的密码不在 VDA 的密码集中，VDA 将拒绝连接。

支持三种密码集：GOV(ernment)、COM(mercial) 和 ALL。 这些密码集中的密码取决于 Windows FIPS 模式；有关 Windows FIPS 模式的信息，请参阅 <http://support.microsoft.com/kb/811833>。 下表列出了受支持的每种密码集中的密码。

SSL 密码集	GOV	COM	ALL	GOV	COM	ALL
FIPS 模式	关	关	关	开	开	开
RSA_KEYX	x	x	x	x	x	x

RSA_SIGN	X	X	X	X	X	X
3DES	X		X	X		X
RC4		X	X			
MD5	X	X	X			
SHA	X	X	X	X	X	X
SHA_256	X	X	X	X	X	X
SHA_384	X	X	X	X	X	X
SHA_512	X	X	X	X	X	X
AES	X	X	X	X	X	X

交付组不能既包含已配置 SSL 的 VDA 又包含未配置 SSL 的 VDA。为交付组配置 SSL 时，应该已经为此交付组中的所有 VDA 配置 SSL。

使用 PowerShell 脚本在 VDA 上配置 SSL

Enable-VdaSSL.ps1 脚本可在 VDA 上启用或禁用 SSL 侦听器。此脚本位于安装介质上的“Support”（支持）>“Tools”（工具）>“SslSupport”文件夹中。

启用 SSL 时，此脚本为指定的 TCP 端口禁用现有的所有 Windows 防火墙规则，然后添加新规则以允许 ICA Service 仅接受 SSL TCP 端口上的传入连接。 它还以下为各项禁用 Windows 防火墙规则：

- Citrix ICA（默认：1494）
- Citrix CGP（默认：2598）
- Citrix WebSocket（默认：8008）

结果，用户仅可以通过 SSL 连接，而不能使用原始 ICA、CGP 或 WebSocket 进行连接。

此脚本包含以下语法描述以及其他示例；您可以使用 Notepad++ 等工具查看此信息。

必须指定 -Enable 或 -Disable 参数；其他所有参数为可选参数。

语法

Enable-VdaSSL {-Enable | -Disable} [-SSLPort <端口>] [-SSLMinVersion "<最低 ssl 版本>"] [-SSLCipherSuite "<密码集>"] [-CertificateThumbPrint "<指纹>"]

参数	说明
-Enable	在 VDA 上安装并启用 SSL 侦听器。 必须带此参数或 -Disable 参数。
-Disable	在 VDA 上禁用 SSL 侦听器。 必须带此参数或 -Enable 参数。 如果指定此参数，其他参数均无效。
-SSLPort <端口>	SSL 端口。 默认值：443
-SSLMinVersion "<最低 ssl 版本>"	最低 SSL 协议版本，两边用引号括起来。 有效值："SSL_3.0"、"TLS_1.0"、"TLS_1.1" 和 "TLS_1.2"。 默认值："TLS_1.0"
-SSLCipherSuite "<密码集>"	SSL 密码集，两边用引号括起来。 有效值："GOV"、"COM" 和 "ALL"。 默认值："ALL"
-CertificateThumbPrint "<指纹>"	证书存储中 SSL 证书的指纹，两边用引号括起来。 通常在证书存储中包含多个证书时使用此参数；脚本使用指纹来选择要使用的证书。 默认值：在证书存储区域的“本地计算机”>“个人”>“证书”区域找到的第一个可用证书。

示例

以下脚本使用所有可选参数的默认值安装并启用 SSL 侦听器。

Enable-VdaSSL -Enable

以下脚本安装并启用 SSL 侦听器，指定 SSL 端口 400、GOV 密码集和最低 TLS 1.2 SSL 协议值。

Enable-VdaSSL -Enable -SSLPort 400 -SSLMinVersion "TLS_1.2" -SSLCipherSuite "GOV"

以下脚本在 VDA 上禁用 SSL 侦听器。

Enable-VdaSSL -Disable

在 VDA 上手动配置 SSL

在 VDA 上手动配置 SSL 时，可以向各个 VDA 上的相应服务授予 SSL 证书私钥的一般读取权限：NT SERVICE\PorticaService（对于 VDA for Windows Desktop OS）或者 NT SERVICE\TermService（对于 VDA for Windows Server OS）。在安装 VDA 的计算机上：

1. 启动 Microsoft 管理控制台 (MMC)：“开始”>“运行”> mmc.exe。
2. 将证书管理单元添加到 MMC：
 1. 选择文件 > 添加/删除管理单元。
 2. 选择证书，然后单击添加。
 3. 收到“该管理单元将始终为下列帐户管理证书”提示时，选择计算机帐户，然后单击下一步。
 4. 收到“请选择需要这个管理单元管理的计算机”提示时，选择本地计算机，然后单击完成。

3. 在“证书(本地计算机)”>“个人”>“证书”下，在证书上单击鼠标右键，然后选择所有任务 > 管理私钥。
4. 访问控制列表编辑器显示“(友好名称)私钥的权限”，其中，(友好名称) 是 SSL 证书的名称。添加以下其中一项服务并向其授予读取权限：
 - 对于 VDA for Windows Desktop OS：“PORTICASERVICE”
 - 对于 VDA for Windows Server OS：“TERMSERVICE”
5. 双击已安装的 SSL 证书。在证书对话框中，选择“详细信息”选项卡，然后滚动到底部。单击指纹。
6. 运行 regedit 并转至 HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\Wds\icawd。
 1. 编辑 SSL 指纹注册表项并将 SSL 证书的指纹值复制到此二进制值中。可以忽略编辑二进制值对话框中的未知项（如 '0000' 和特殊字符），这样是安全的。
 2. 编辑 SSLEnabled 注册表项并将 DWORD 值更改为 1。（之后要禁用 SSL，请将 DWORD 值更改为 0。）
 3. 如果要更改默认设置（可选），请在相同注册表路径中使用以下值：
 - SSLPort DWORD – SSL 端口号。默认值：443。
 - SSLMinVersion DWORD – 1 = SSL 3.0、2 = TLS 1.0、3 = TLS 1.1、4 = TLS 1.2。默认值：2 (TLS 1.0)。
 - SSLCipherSuite DWORD – 1 = GOV、2 = COM、3 = ALL。默认值：3 (ALL)。
7. 如果 SSL TCP 端口不是默认值 443，请确保此端口在 Windows 防火墙中处于打开状态。（在 Windows 防火墙中创建入站规则时，请确保其属性已选中“允许连接”或“启用”条目）。
8. 确保没有其他应用程序或服务（如 IIS）正在使用 SSL TCP 端口。
9. 对于 VDA for Windows Server OS，重新启动计算机以使更改生效。（无需重新启动包含 VDA for Windows Desktop OS 的计算机）。

在交付组上配置 SSL

为包含已配置 SSL 连接的 VDA 的每个交付组完成此过程。

1. 从 Studio，打开 PowerShell 控制台。
2. 运行 asnp Citrix.* 以加载 Citrix 产品 cmdlet。
3. 运行 Get-BrokerAccessPolicyRule -DesktopGroupName '<交付组名称>' | Set-BrokerAccessPolicyRule -HdxSslEnabled \$true。
其中，<交付组名称> 是包含 VDA 的交付组的名称。
4. 运行 Set-BrokerSite -DnsResolutionEnabled \$true。

故障排除

如果连接出错，请检查 VDA 的系统事件日志。

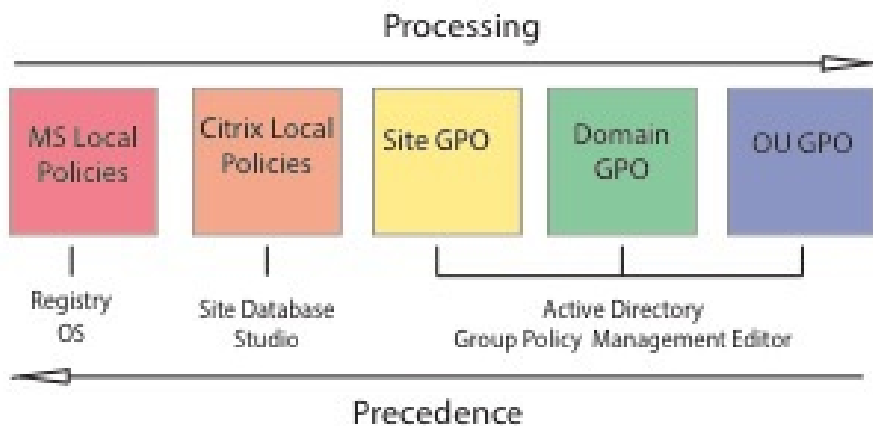
使用 Receiver for Windows 时，如果收到指示 SSL 错误的连接错误（如 1030），请禁用 Desktop Viewer，然后尝试重新连接。尽管连接仍会失败，但可能会提供对基本 SSL 问题的解释（例如，在从证书颁发机构申请证书时指定了错误的模板）。

策略

May 04, 2016

策略是设置的集合，这些设置定义如何为一组用户、设备或连接类型管理会话、带宽和安全性。

可以为物理计算机和虚拟机或用户提供策略设置。可以向本地级别或 Active Directory 的安全组中的单个用户应用设置。配置定义具体的条件和规则，如果您没有明确分配策略，设置将应用于所有连接。



可以在网络的不同级别应用策略。位于组织单位 GPO 级别的策略设置在网络上具有最高优先级。域 GPO 级别的策略覆盖站点组策略对象级别的策略，后者覆盖 Microsoft 和 Citrix 本地策略级别上任何存在冲突的策略。

所有 Citrix 本地策略在 Citrix Studio 控制台上创建和管理，并存储在站点数据库中；而组策略使用 Microsoft 组策略管理控制台 (GPMC) 进行创建和管理，并存储在 Active Directory 中。Microsoft 本地策略在 Windows 操作系统中创建，存储在注册表中。

Studio 使用建模向导帮助管理员比较模板和策略中的配置设置，以便排除冲突和冗余设置。管理员可以使用 GPMC 设置 GPO 以配置设置并将其应用于网络上不同级别的目标用户集合中。

这些 GPO 保存在 Active Directory 中，大多数 IT 通常会限制对这些设置管理的访问以确保安全。

设置根据优先级及其条件合并。任何禁用设置都会覆盖等级较低的启用设置。未配置的策略设置会被忽略，且不会覆盖等级较低的设置。

本地策略也可能会与 Active Directory 中的组策略冲突，在这种情况下，二者会根据具体情况相互覆盖。

所有策略按照以下顺序处理：

1. 最终用户使用域凭据登录计算机。
2. 凭据被发送到域控制器。
3. Active Directory 应用所有策略（最终用户、端点、组织单位和域）。
4. 最终用户登录 Receiver 并访问应用程序或桌面。
5. 为最终用户和托管资源的计算机处理 Citrix 和 Microsoft 策略。
6. Active Directory 确定策略设置的优先级，并将其应用于端点设备的注册表和托管资源的计算机。
7. 最终用户从资源注销。最终用户和端点设备的 Citrix 策略不再起作用。
8. 最终用户注销用户设备，从而释放 GPO 用户策略。
9. 最终用户关闭设备，从而释放 GPO 计算机策略。

为一组用户、设备和计算机创建策略时，有些成员可能会有其他要求，并可能需要设置某些策略设置的例外情况。例外通过 Studio 和 GPMC 中的过滤器方式实现，用以确定策略所影响的人员或内容。

相关内容

- [使用策略](#)
- [策略模板](#)
- [创建策略](#)
- [对策略进行比较、设定优先级、建模和故障排除](#)
- [默认策略设置](#)
- [策略设置参考](#)

使用策略

May 04, 2016

配置 Citrix 策略以控制用户访问或会话环境。Citrix 策略是控制连接、安全性和带宽设置最有效的方法。您可以针对特定用户组、设备或连接类型创建策略。每个策略可以包含多个设置。

处理 Citrix 策略的工具

可以使用以下工具处理 Citrix 策略。

- **Studio** - 如果您是 Citrix 管理员，但没有管理组策略的权限，可以使用 Studio 为您的站点创建策略。使用 Studio 创建的策略存储在站点数据库中，当虚拟桌面注册到代理或用户连接到虚拟桌面时，系统会将更新信息推送到该虚拟桌面。
- **本地组策略编辑器** (Microsoft 管理控制台管理单元) - 如果您的网络环境使用 Active Directory，并且您拥有管理组策略的权限，则可以使用本地组策略编辑器为站点创建策略。您配置的设置会对在组策略管理控制台中指定的组策略对象 (GPO) 产生影响。
重要：必须使用本地组策略编辑器配置某些策略设置，包括与向 Controller 注册 VDA 相关的策略设置和与 App-V 服务器相关的策略设置。

策略处理顺序和优先级

组策略设置的处理顺序如下：

1. 本地 GPO
2. XenApp 或 XenDesktop 站点 GPO（存储在站点数据库中）
3. 站点级 GPO
4. 域级 GPO
5. 组织单位

但是，如果存在冲突，最后处理的策略设置会覆盖较早处理的策略设置。这意味着策略设置的优先级顺序如下：

1. 组织单位
2. 域级 GPO
3. 站点级 GPO
4. XenApp 或 XenDesktop 站点 GPO（存储在站点数据库中）
5. 本地 GPO

例如，Citrix 管理员使用 Studio 创建了一个策略（策略 A），用于为公司的销售员工启用客户端文件重定向。同时，另一名管理员使用组策略编辑器也创建了一个策略（策略 B），用于为销售员工禁用客户端文件重定向。当销售员工登录到虚拟桌面时，将应用策略 B 而忽略策略 A，因为策略 B 在域级别处理，而策略 A 在 XenApp 或 XenDesktop 站点 GPO 级别处理。

但是，当用户启动 ICA 或远程桌面协议 (RDP) 会话时，Citrix 会话设置将覆盖在 Active Directory 策略中或使用远程桌面会话主机配置进行配置的设置。这包括与典型的 RDP 客户端连接设置相关的设置（例如桌面墙纸、菜单动画以及拖动时查看窗口内容）。

使用多个策略时，可以向包含冲突设置的策略分配优先级，请参阅[对策略进行比较、设定优先级、建模和故障排除](#)了解详细信息。

Citrix 策略工作流

策略的配置过程如下：

1. 创建策略。
2. 配置策略设置。

3. 将策略分配给计算机和用户对象。
4. 设定策略的优先级。
5. 通过运行 Citrix 组策略建模向导确认有效策略。

导航 Citrix 策略和设置

在本地组策略编辑器中，策略和设置分为两个类别显示：计算机配置和用户配置。每个类别都具有 Citrix 策略节点。请参阅 Microsoft 文档以了解导航和使用此管理单元的详细信息。

在 Studio 中，策略设置按其所影响的功能分为多个类别。例如，Profile Management 部分包含用于 Profile Management 的策略设置。

- 计算机设置（应用于计算机的策略设置）定义虚拟桌面的行为并在虚拟桌面启动时应用。即使虚拟桌面上没有活动的用户会话，也会应用这些设置。用户设置定义了使用 ICA 连接时的用户体验。当用户使用 ICA 连接或重新连接时应用用户策略。如果用户使用 RDP 连接或直接登录控制台，将不应用用户策略。

要访问策略、设置或模板，请在 Studio 导航窗格中选择策略。

- **策略**选项卡列出所有策略。选择某个策略时，右侧的选项卡显示：概览（名称、优先级、启用/禁用状态和说明）、设置（已配置设置的列表）和已分配给（策略当前分配到的用户和计算机对象）。有关详细信息，请参阅[创建策略](#)。
- **模板**选项卡列出 Citrix 提供的模板和您创建的自定义模板。选择模板时，右侧的选项卡显示：说明（要使用此模板的原因）和设置（已配置设置的列表）。有关详细信息，请参阅[策略模板](#)。
- 利用**比较**选项卡，您可以将某个策略或模板中的设置与其他策略或模板中的设置进行比较。例如，您可能希望验证设置值以确保符合最佳实践。有关详细信息，请参阅[对策略进行比较、设定优先级、建模和故障排除](#)。
- 从**建模**选项卡，可以利用 Citrix 策略模拟连接场景。有关详细信息，请参阅[对策略进行比较、设定优先级、建模和故障排除](#)。

要搜索策略或模板中的设置，请执行以下操作：

1. 选择策略或模板。
2. 在“操作”窗格中选择编辑策略或编辑模板。
3. 在设置页面，首先输入设置的名称。

可以通过选择特定的产品版本或类别（例如带宽），或通过选中仅查看所选对象复选框，或选择仅搜索已添加到选定策略中的设置，来精简搜索结果。对于未过滤的搜索，请选择所有设置。

- 要在策略中搜索设置，请执行以下操作：
 1. 选择该策略。
 2. 选择设置选项卡，首先输入设置的名称。

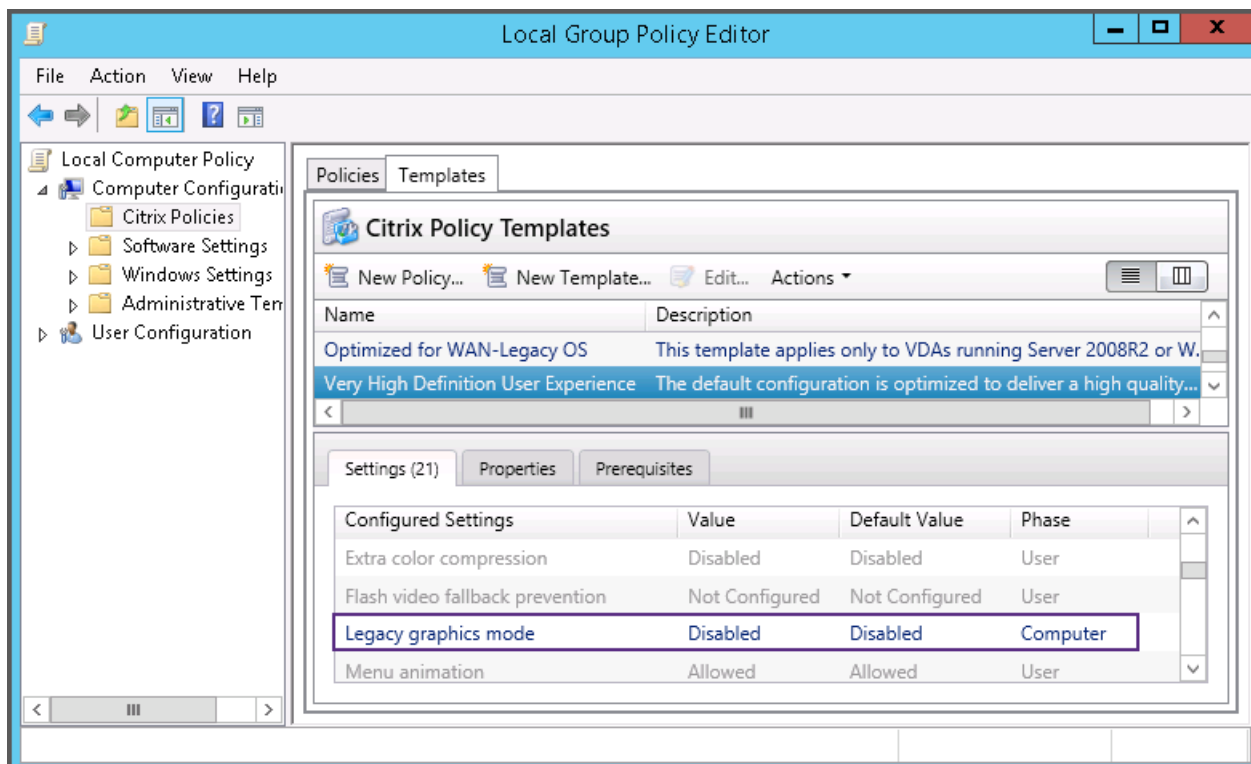
可以通过选择特定产品版本或类别精简搜索结果。对于未过滤的搜索，请选择所有设置。

策略一旦创建，便完全独立于所使用的模板。您可以使用新策略的“说明”字段来跟踪所使用的源模板。

在 Studio 中，策略和模板显示在单个列表中，不管它们是否包含用户、计算机或两种设置类型，并且可以使用用户和计算机过滤器进行应用。

在组策略编辑器中，计算机和用户设置必须单独应用，即使是通过包含两种设置类型的模板创建也是如此。在此示例中，选择使用计算机配置中的超高清清晰度用户体验：

- 旧图形模式是用于通过此模板创建的策略的计算机设置。
- 灰显的用户设置不用于通过此模板创建的策略。



策略模板

May 04, 2016

模板是从预定义的起点创建策略的源。内置 Citrix 模板已针对特定环境或网络条件优化，可以用作：

- 用于创建自己的策略和模板以在站点之间共享的源。
- 易于在部署之间比较结果的参考，因为您将可以在结果两边加上引号，例如，“..when using Citrix template x or y..”。
- 通过导入或导出模板与 Citrix 支持或可信第三方通信策略的方法。

策略模板可以导入或导出。有关其他模板和内置模板的更新，请参阅 [CTX202000](#)。

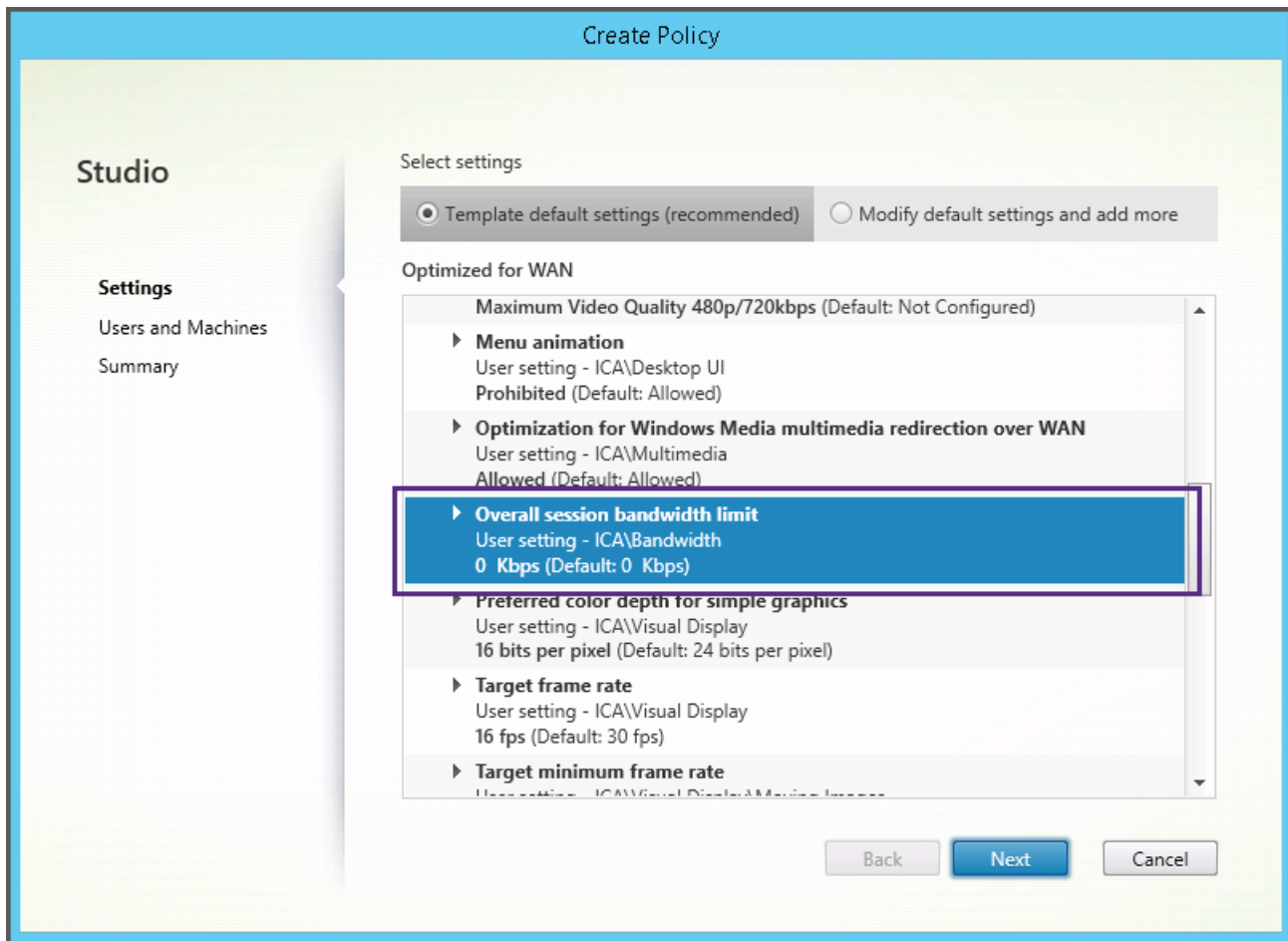
有关使用模板创建策略的注意事项，请参阅 [CTX202330](#)。

7.6 FP3 中的内置 Citrix 模板

XenApp 和 XenDesktop 7.6 FP3 组策略管理软件包 (2.5.0.0) 包括以下策略模板，这些模板取代且增强了之前提供的内置 Citrix 模板：

- **超高清用户体验。**此模板强制实施尽可能实现最佳用户体验的默认设置。在按照优先级顺序处理多个策略的场景中使用此模板。
- **高服务器可扩展性。**应用此模板可节约服务器资源。此模板可以平衡用户体验和服务器可扩展性。它可以提供良好的用户体验，同时增加单个服务器上可以托管的用户数。此模板不使用视频编解码器压缩图像，并阻止服务器端进行多媒体呈现。
- **高服务器可扩展性-旧版操作系统。**此高服务器可扩展性模板仅适用于运行 Server 2008 R2 或 Windows 7 及更早版本的 VDA。此模板依靠对这些操作系统较为有效的旧图形模式。
- **WAN 优化。**此模板旨在用于满足以下条件的任务型工作人员：位于使用共享 WAN 连接的分支结构或采用低带宽连接的远程位置，访问具有简单图形用户界面且包含很少多媒体内容的应用程序。此模板通过降低视频回退体验和某些服务器可扩展性实现最佳带宽效率。
- **WAN 优化-旧版操作系统。**此 WAN 优化模板仅适用于运行 Server 2008 R2 或 Windows 7 及更早版本的 VDA。此模板依靠对这些操作系统较为有效的旧图形模式。
- **安全性与控制。**在低风险容忍的环境中使用此模板，可尽量减少 XenApp 和 XenDesktop 中默认启用的功能。此模板中包含的设置可禁止在用户设备上访问打印、剪贴板、外围设备、驱动器映射、端口重定向以及 Flash 加速。应用此模板可能会占用更多带宽并降低每个服务器的用户密度。

尽管我们建议使用内置 Citrix 模板的默认设置，但是您会发现有些设置没有具体的建议值。例如，WAN 优化模板中的“总会话带宽限制”。在这种情况下，模板采用展示设置的方法，以便管理员了解此设置可能适用的情况。



如果您使用 XenApp 和 XenDesktop 7.6FP3 之前的部署（策略管理和 VDA）并需要高服务器可扩展性和 WAN 优化模板，请在应用时使用这些模板的旧版操作系统版本。

注意

内置模板由 Citrix 创建，并由 Citrix 负责更新。您无法修改或删除这些模板。

使用 Studio 创建和管理模板

基于模板创建新模板：

1. 在 Studio 导航窗格中选择**策略**。
2. 选择**模板**选项卡，然后选择创建新模板时要基于的模板。
3. 在“操作”窗格中选择**创建模板**。
4. 选择并配置要包含在新模板中的策略设置。删除不应包含的所有现有设置。为新模板输入一个名称。

单击**完成**后，新模板显示在**模板**选项卡中。

基于策略创建新模板：

1. 在 Studio 导航窗格中选择**策略**。
2. 选择**策略**选项卡，然后选择创建新模板时要基于的策略。

- 3. 在“操作”窗格中选择**另存为模板**。
- 4. 选择并配置要包含在模板中的任何新策略设置。删除不应包含的所有现有设置。输入模板的名称和说明，然后单击**完成**。

导入模板：

- 1. 在 Studio 导航窗格中选择**策略**。
- 2. 选择**模板**选项卡，然后选择**导入模板**。
- 3. 选择要导入的模板文件，然后单击**打开**。如果您要导入的模板与某个现有模板同名，则可以选择覆盖现有模板，或使用自动生成的其他名称保存该模板。

导出模板：

- 1. 在 Studio 导航窗格中选择**策略**。
- 2. 选择**模板**选项卡，然后选择**导出模板**。
- 3. 选择模板的保存位置，然后单击**保存**。

将在指定位置创建一个 .gpt 文件。

使用组策略编辑器创建和管理模板

从组策略编辑器，展开计算机配置或用户配置。展开策略节点，然后选择 Citrix 策略。从下面选择合适的操作。

任务	说明
基于现有策略创建新模板	在“策略”选项卡上，选择策略，然后选择操作 > 另存为模板。
基于现有模板创建新策略	在“模板”选项卡上，选择模板，然后单击新建策略。
基于现有模板创建新模板	在“模板”选项卡上，选择模板，然后单击新建模板。
导入模板	在“模板”选项卡上，选择操作 > 导入。
导出模板	在“模板”选项卡上，选择操作 > 导出。
查看模板设置	在“模板”选项卡上，选择模板，然后单击“设置”选项卡。
查看模板属性的摘要	在“模板”选项卡上，选择模板，然后单击“属性”选项卡。
查看模板必备项	在“模板”选项卡上，选择模板，然后单击“必备项”选项卡。

模板和委派管理

策略模板存储在安装策略管理软件包的计算机上。此计算机为 Delivery Controller 计算机或组策略对象管理计算机，而不是 XenApp 和 XenDesktop 站点的数据库。这意味着策略模板文件由 Windows 管理权限（而非站点的委派管理角色和作用域）控制。

因此，站点中具有只读权限的管理员可以创建新模板。但是，因为模板是本地文件，实际上并未对环境进行任何更改。

自定义模板仅对创建了这些模板的用户帐户可见，并且存储在用户的 Windows 配置文件中。要进一步显示自定义模板，请基于该模板创建一条策略，或将其导出到共享位置。

创建策略

May 04, 2016

创建策略前，确定将受此策略影响的用户或设备组。您可能希望基于用户工作职责、连接类型、用户设备或地理位置来创建策略。也可以使用用于 Windows Active Directory 组策略的条件。

如果已创建应用于某个组的策略，请考虑编辑该策略并配置适当的设置，而不是创建另一个策略。请避免单纯为了启用特定设置或拒绝将该策略应用到特定用户而创建新策略。

创建新策略时，可以将策略模板中的设置作为基础并根据需要自定义设置，也可以不使用模板，然后添加所需的全部设置。

策略设置

策略设置的状态可以是已启用、已禁用或未配置。默认情况下，策略设置的状态是未配置，表示未将其添加到策略。仅在将设置添加到策略后才应用这些设置。

某些策略设置可处于以下状态之一：

- 允许或禁止，允许或阻止由设置控制的操作。在某些情况下，会允许或阻止用户在会话中管理设置的操作。例如，如果菜单动画设置为“允许”，则用户可以在其客户端环境中控制菜单动画。
- 启用或禁用，打开或关闭设置。如果禁用了某个设置，则在任何等级较低的策略中都不会启用该设置。

此外，某些设置还可控制相关设置的效果。例如，客户端驱动器重定向设置控制是否允许用户访问其设备上的驱动器。要允许用户访问其网络驱动器，必须同时将此设置和客户端网络驱动器设置添加到策略中。如果客户端驱动器重定向设置处于禁用状态，用户将无法访问其网络驱动器，即使客户端网络驱动器设置处于启用状态也是如此。

通常，影响计算机的策略设置更改会在虚拟桌面重新启动时或用户登录时生效。影响用户的策略设置更改会在用户下次登录时生效。如果您使用的是 Active Directory，策略设置会在 Active Directory 每隔 90 分钟定期重新评估策略时更新，并在虚拟桌面重新启动时或用户登录时应用。

对于某些策略设置，可在将设置添加到策略时输入或选择一个值。可以通过选择使用默认值限制对设置进行配置；这样可禁止对设置进行配置，在应用策略时仅允许使用设置的默认值，不考虑在选择使用默认值之前输入的值。

最佳做法：

- 将策略分配给组，而非单个用户。如果将策略分配给组，则将用户添加到组或从组中删除用户时，分配的策略会自动更新。
- 请勿启用“远程桌面会话主机配置”中的冲突或重叠设置。在某些情况下，“远程桌面会话主机配置”可提供与 Citrix 策略设置相似的功能。如有可能，请将所有设置的状态保持一致（已启用或已禁用），以便进行故障排除。
- 禁用未使用的策略。未添加任何设置的策略会带来不必要的处理过程。

策略分配

创建策略时，将其分配给某些用户和计算机对象；根据特定条件或规则将该策略应用于连接。一般情况下，可以根据条件组合向策略添加任意数量的分配。如果指定不进行分配，策略将应用于所有连接。

下表列出了可用的分配：

分配名称	应用策略的根据
访问控制	连接客户端所依据的访问控制条件 ● 连接类型 - 将策略应用到使用 NetScaler Gateway 建立的连接还是不使用 NetScaler Gateway 建立的连接。 ● NetScaler Gateway 场名称 - NetScaler Gateway 虚拟服务器的名称。

分配名称	访问条件。要使用的终端分析策略或会话策略的名称。 应用策略的根据
Citrix CloudBridge	是否通过 Citrix CloudBridge 启动用户会话。 注意：您只能向策略中添加一个 Citrix CloudBridge 分配。
客户端 IP 地址	用于连接到会话的用户设备的 IP 地址。 - IPv4 示例：12.0.0.0、12.0.0.*、12.0.0.1-12.0.0.70、12.0.0.1/24 - IPv6 示例：2001:0db8:3c4d:0015:0:0:abcd:ef12, 2001:0db8:3c4d:0015::/54
客户端名称	用户设备的名称 - 精确匹配：ClientABCName - 使用通配符：Client*Name
交付组	交付组成员身份。
交付组类型	桌面或应用程序的类型：专用桌面、共享桌面、专用应用程序或共享应用程序。
组织单位 (OU)	组织单位。
标记	标记。 注意：为确保在使用标记时正确应用策略，请安装 CTX142439 中的修补程序。
用户或组	用户或组名称。

用户登录时，系统会确定与连接的分配相匹配的所有策略。这些策略按优先级排序，并对任意设置的多个实例进行比较。根据策略的优先级应用每个设置。任何已禁用的策略设置都优先于级别较低的已启用规则。未配置的策略设置会被忽略。

重要：如果使用组策略管理控制台同时配置 Active Directory 和 Citrix 策略，可能无法按预期应用分配和设置。有关详细信息，请参阅 [CTX127461](#)。

默认情况下，提供名为“未过滤”的策略。

- 如果使用 Studio 来管理 Citrix 策略，添加到“未过滤”策略的设置将应用到站点中的所有服务器、桌面和连接。
- 如果使用本地组策略编辑器管理 Citrix 策略，添加到“未过滤”策略的设置将应用到包含该策略的组策略对象 (GPO) 作用域内的所有站点和连接。例如，Sales 销售团队包含名为 Sales-US 的组策略对象，该组策略对象包含美国销售团队的所有成员。该 Sales-US 组策略对象是使用包含多项用户策略设置的“未过滤”策略进行配置的。美国的销售经理登录到站点时，“未过滤”策略中的设置会自动应用到会话，因为该用户属于 Sales-US 组策略对象的成员。

分配的模式决定策略是否仅应用到符合所有分配条件的连接。如果将模式设置为允许（默认设置），策略将仅应用到符合分配条件的连接。如果将模式设置为拒绝，将在连接不符合分配条件时应用策略。下例说明了存在多个分配时分配模式对 Citrix 策略的影响。

- **示例：模式不同但类型相同的分配** - 如果策略中包含两个类型相同的分配，其中一个设置为允许，一个设置为拒绝，假设连接同时满足这两个分配，则设置为拒绝的分配优先级较高。例如：
策略 1 包含以下分配：

- 分配 A 指定销售组，且模式设置为允许
- 分配 B 指定销售经理的帐户，且模式设置为拒绝

由于分配 B 的模式设置为拒绝，因此即使销售经理属于销售组，在他登录到站点时也不会应用该策略。

- **示例：模式相同但类型不同的分配** - 在包含两个或更多个类型不同但模式设置为允许的策略中，连接必须至少满足每种类型的一个分配，才能应用该策略。例如：

策略 2 包含以下分配：

- 分配 C 为用户分配，用于指定销售组，且模式设置为允许
 - 分配 D 为客户端 IP 地址分配，用于指定 10.8.169.*（企业网络），且模式设置为允许
- 销售经理从办公室登录到站点时，会应用该策略，因为连接同时满足这两个分配。

策略 3 包含以下分配：

- 分配 E 为用户分配，用于指定销售组，且模式设置为允许
 - 分配 F 为访问控制分配，用于指定 NetScaler Gateway 连接条件，且模式设置为允许
- 销售经理从办公室登录到站点时，不会应用该策略，因为连接不满足分配 F。

使用 Studio 基于模板创建新策略

1. 在 Studio 导航窗格中选择策略。
2. 选择“模板”选项卡，然后选择一个模板。
3. 在“操作”窗格中选择基于模板创建策略。
4. 默认情况下，新策略使用模板中的所有默认设置（选中使用模板默认设置单选按钮）。如果要更改设置，请选中修改默认值并添加更多设置单选按钮，然后添加或删除设置。
5. 通过选择以下其中一个选项指定应用策略的方式：
 - 分配给所选用户和计算机对象并选择要应用策略的用户和计算机对象。
 - 分配给站点中的所有对象以将策略应用到站点中的所有用户和计算机对象。
6. 输入策略的名称（或接收默认名称）；考虑根据受影响的用户或对象来命名策略，例如，Accounting Department 或 Remote Users。提供说明（可选）。

策略在默认情况下启用，您可以将其禁用。启用策略将使策略立即应用到登录的用户。禁用策略可阻止应用策略。如果您过后需要设定策略的优先级或添加设置，请考虑禁用策略，直至准备好应用此策略。

使用 Studio 创建新策略

1. 在 Studio 导航窗格中选择策略。
2. 选择策略选项卡。
3. 在“操作”窗格中选择创建策略。
4. 添加并配置策略设置。
5. 通过选择以下其中一个选项指定应用策略的方式：
 - 分配给所选用户和计算机对象并选择要应用策略的用户和计算机对象。
 - 分配给站点中的所有对象以将策略应用到站点中的所有用户和计算机对象。
6. 输入策略的名称（或接收默认名称）；考虑根据受影响的用户或对象来命名策略，例如，Accounting Department 或 Remote Users。提供说明（可选）。

策略在默认情况下启用，您可以将其禁用。启用策略将使策略立即应用到登录的用户。禁用策略可阻止应用策略。如果您过后需要设定策略的优先级或添加设置，请考虑禁用策略，直至准备好应用此策略。

使用组策略编辑器创建和管理策略

从组策略编辑器，展开计算机配置或用户配置。展开策略节点，然后选择 Citrix 策略。从下面选择合适的操作。

任务	说明

创建新策略	在“策略”选项卡上，单击新建。
编辑现有策略	在“策略”选项卡上，选择策略，然后单击编辑。
更改现有策略的优先级	在“策略”选项卡上，选择策略，然后单击提高或降低。
查看策略的摘要信息	在“策略”选项卡上，选择策略，然后单击“摘要”选项卡。
查看和修改策略设置	在“策略”选项卡上，选择策略，然后单击“设置”选项卡。
查看和修改策略过滤器	在“策略”选项卡上，选择策略，然后单击“过滤器”选项卡。
启用或禁用策略	在“策略”选项卡上，选择策略，然后选择操作 > 启用或操作 > 禁用。
基于现有模板创建新策略	在“模板”选项卡上，选择模板，然后单击新建策略。

对策略进行比较、设定优先级、建模和故障排除

Nov 02, 2016

You can use multiple policies to customize your environment to meet users' needs based on their job functions, geographic locations, or connection types. For example, for security you may need to place restrictions on user groups who regularly work with sensitive data. You can create a policy that prevents users from saving sensitive files on their local client drives. However, if some people in the user group do need access to their local drives, you can create another policy for only those users. You then rank or prioritize the two policies to control which one takes precedence.

When using multiple policies, you must determine how to prioritize them, how to create exceptions, and how to view the effective policy when policies conflict.

In general, policies override similar settings configured for the entire Site, for specific Delivery Controllers, or on the user device. The exception to this principle is security. The highest encryption setting in your environment, including the operating system and the most restrictive shadowing setting, always overrides other settings and policies.

Citrix policies interact with policies you set in your operating system. In a Citrix environment, Citrix settings override the same settings configured in an Active Directory policy or using Remote Desktop Session Host Configuration. This includes settings that are related to typical Remote Desktop Protocol (RDP) client connection settings such as Desktop wallpaper, Menu animation, and View window contents while dragging. For some policy settings, such as Secure ICA, the settings in policies must match the settings in the operating system. If a higher priority encryption level is set elsewhere, the Secure ICA policy settings that you specify in the policy or when you are delivering application and desktops can be overridden.

For example, the encryption settings that you specify when creating Delivery Groups should be at the same level as the encryption settings you specified throughout your environment.

Note: In the second hop of double-hop scenarios, when a Desktop OS VDA connects to Server OS VDA, Citrix policies act on the Desktop OS VDA as if it were the user device. For example, if policies are set to cache images on the user device, the images cached for the second hop in a double-hop scenario are cached on the Desktop OS VDA machine.

Compare policies and templates

You can compare settings in a policy or template with those in other policies or templates. For example, you might need to verify setting values to ensure compliance with best practices. You might also want to compare settings in a policy or template with the default settings provided by Citrix.

1. Select Policies in the Studio navigation pane.
2. Click the Comparison tab and then click Select.
3. Choose the policies or templates to compare. To include default values in the comparison, select the Compare to default settings check box.
4. After you click Compare, the configured settings are displayed in columns.
5. To see all settings, select Show All Settings. To return to the default view, select Show Common Settings.

Prioritize policies

Prioritizing policies allows you to define the precedence of policies when they contain conflicting settings. When a user logs on, all policies that match the assignments for the connection are identified. Those policies are sorted into priority order and multiple instances of any setting are compared. Each setting is applied according to the priority ranking of the policy.

You prioritize policies by giving them different priority numbers in Studio. By default, new policies are given the lowest priority. If policy settings conflict, a policy with a higher priority (a priority number of 1 is the highest) overrides a policy with a

lower priority. Settings are merged according to priority and the setting's condition; for example, whether the setting is disabled or enabled. Any disabled setting overrides a lower-ranked setting that is enabled. Policy settings that are not configured are ignored and do not override the settings of lower-ranked settings.

1. Select Policies in the Studio navigation pane. Make sure the Policies tab is selected.
2. Select a policy.
3. Select Lower Priority or Higher Priority in the Actions pane.

Exceptions

When you create policies for groups of users, user devices, or machines, you may find that some members of the group require exceptions to some policy settings. You can create exceptions by:

- Creating a policy only for those group members who need the exceptions and then ranking the policy higher than the policy for the entire group
- Using the Deny mode for an assignment added to the policy

An assignment with the mode set to Deny applies a policy only to connections that do not match the assignment criteria. For example, a policy contains the following assignments:

- Assignment A is a client IP address assignment that specifies the range 208.77.88.*; the mode is set to Allow
- Assignment B is a user assignment that specifies a particular user account; the mode is set to Deny

The policy is applied to all users who log on to the Site with IP addresses in the range specified in Assignment A. However, the policy is not applied to the user logging on to the Site with the user account specified in Assignment B, even though the user's computer is assigned an IP address in the range specified in Assignment A.

Determine which policies apply to a connection

Sometimes a connection does not respond as expected because multiple policies apply. If a higher priority policy applies to a connection, it can override the settings you configure in the original policy. You can determine how final policy settings are merged for a connection by calculating the Resultant Set of Policy.

You can calculate the Resultant Set of Policy in the following ways:

- Use the Citrix Group Policy Modeling Wizard to simulate a connection scenario and discern how Citrix policies might be applied. You can specify conditions for a connection scenario such as domain controller, users, Citrix policy assignment evidence values, and simulated environment settings such as slow network connection. The report that the wizard produces lists the Citrix policies that would likely take effect in the scenario. If you are logged on to the Controller as a domain user, the wizard calculates the Resultant Set of Policy using both site policy settings and Active Directory Group Policy Objects (GPOs).
- Use Group Policy Results to produce a report describing the Citrix policies in effect for a given user and controller. The Group Policy Results tool helps you evaluate the current state of GPOs in your environment and generates a report that describes how these objects, including Citrix policies, are currently being applied to a particular user and controller.

You can launch the Citrix Group Policy Modeling Wizard from the Actions pane in Studio. You can launch either tool from the Group Policy Management Console in Windows.

If you run the Citrix Group Policy Modeling Wizard or Group Policy Results tool from the Group Policy Management Console, site policy settings created using Studio are not included in the Resultant Set of Policy.

To ensure you obtain the most comprehensive Resultant Set of Policy, Citrix recommends launching the Citrix Group Policy Modeling wizard from Studio, unless you create policies using only the Group Policy Management Console.

Use the Citrix Group Policy Modeling Wizard

Open the Citrix Group Policy Modeling Wizard using one of the following:

- Select Policies in the Studio navigation pane, select the Modeling tab, and then select Launch Modeling Wizard in the Actions pane.
- Launch the Group Policy Management Console (gpmc.msc), right-click Citrix Group Policy Modeling in the tree pane, and then select Citrix Group Policy Modeling Wizard.

Follow the wizard instructions to select the domain controller, users, computers, environment settings, and Citrix assignment criteria to use in the simulation. After you click Finish, the wizard produces a report of the modeling results. In Studio, the report appears in the middle pane under the Modeling tab.

To view the report, select View Modeling Report.

Troubleshoot policies

Users, IP addresses, and other assigned objects can have multiple policies that apply simultaneously. This can result in conflicts where a policy may not behave as expected. When you run the Citrix Group Policy Modeling Wizard or the Group Policy Results tool, you might discover that no policies are applied to user connections. When this happens, users connecting to their applications and desktops under conditions that match the policy evaluation criteria are not affected by any policy settings. This occurs when:

- No policies have assignments that match the policy evaluation criteria.
- Policies that match the assignment do not have any settings configured.
- Policies that match the assignment are disabled.

If you want to apply policy settings to the connections that meet the specified criteria, make sure:

- The policies you want to apply to those connections are enabled.
- The policies you want to apply have the appropriate settings configured.

默认策略设置

Nov 02, 2016

The following tables list policy settings, their default, and the Virtual Delivery Agent (VDA) versions to which they apply.

ICA

Name	Default setting	VDA
Client clipboard redirection	Allowed	All VDA versions
Desktop launches	Prohibited	VDA for Server OS 7 through current VDA for Server OS
ICA listener connection timeout	120000 milliseconds	VDA 5, 5.5, 5.6 Feature Pack 1, VDA for Desktop OS 7 through current VDA for Desktop OS
ICA listener port number	1494	All VDA versions
Launching of non-published programs during client connection	Prohibited	VDA for Server OS 7 through current VDA for Server OS
Client clipboard write allowed formats	No formats are specified	VDA 7.6
Restrict client clipboard write	Prohibited	VDA 7.6
Restrict session clipboard write	Prohibited	VDA 7.6
Session clipboard write allowed formats	No formats are specified	VDA 7.6

ICA/Adobe Flash Delivery/Flash Redirection

Name	Default setting	VDA
Flash video fallback prevention	Not configured	VDA 7.6 FP3
Flash video fallback prevention error *.swf		VDA 7.6 FP3

ICA/Audio

Name	Default setting	VDA

Name	Default setting	VDA
Audio Plug N Play	Allowed	VDA for Server OS 7 through current VDA for Server OS
Audio quality	High - high definition audio	All VDA versions
Client audio redirection	Allowed	All VDA versions
Client microphone redirection	Allowed	All VDA versions

ICA/Auto Client Reconnect

Name	Default setting	VDA
Auto client reconnect	Allowed	VDA
Auto client reconnect authentication	Do not require authentication	VDA
Auto client reconnect logging	Do not log auto-reconnect events	VDA

ICA/Bandwidth

Name	Default setting	VDA
Audio redirection bandwidth limit	0 Kbps	VDA
Audio redirection bandwidth limit percent	0	VDA
Client USB device redirection bandwidth limit	0 Kbps	VDA 5.5, 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Client USB device redirection bandwidth limit percent	0	VDA 5.5, 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Clipboard redirection bandwidth limit	0 Kbps	All VDA versions
Clipboard redirection bandwidth limit percent	0	All VDA versions
COM port redirection bandwidth limit	0 Kbps	All VDA versions; for VDA 7.x, configure this setting using the registry.
COM port redirection bandwidth limit percent	0	All VDA versions; for VDA 7.x, configure this setting using the registry.

Name	Default setting	VDA
File redirection bandwidth limit	0 Kbps	All VDA versions
File redirection bandwidth limit percent	0	All VDA versions
HDX MediaStream Multimedia Acceleration bandwidth limit	0 Kbps	VDA 5.5, 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
HDX MediaStream Multimedia Acceleration bandwidth limit percent	0	VDA 5.5, 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
LPT port redirection bandwidth limit	0 Kbps	All VDA versions; for VDA 7.x, configure this setting using the registry.
LPT port redirection bandwidth limit percent	0	All VDA versions; for VDA 7.x, configure this setting using the registry.
Overall session bandwidth limit	0 Kbps	All VDA versions
Printer redirection bandwidth limit	0 Kbps	All VDA versions
Printer redirection bandwidth limit percent	0	All VDA versions
TWAIN device redirection bandwidth limit	0 Kbps	VDA 5.5, 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
TWAIN device redirection bandwidth limit percent	0	VDA 5.5, 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS

ICA/Client Sensors

Name	Default setting	VDA
Allow applications to use the physical location of the client device	Prohibited	VDA 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS

ICA/Desktop UI

Name	Default setting	VDA
Desktop Composition Redirection	Disabled (7.6 FP3 through current) Enabled (5.6 through 7.6)	VDA 5.6, VDA for Desktop OS 7 through current, VDA

Name	EP2) Default setting	VDA
Desktop Composition Redirection graphics quality	Medium	VDA 5.6, VDA for Desktop OS 7 through current, VDA
Desktop wallpaper	Allowed	All VDA versions
Menu animation	Allowed	All VDA versions
View window contents while dragging	Allowed	All VDA versions

ICA/End User Monitoring

Name	Default setting	VDA
ICA round trip calculation	Enabled	All VDA versions
ICA round trip calculation interval	15 seconds	All VDA versions
ICA round trip calculations for idle connections	Disabled	All VDA versions

ICA/Enhanced Desktop Experience

Name	Default setting	VDA
Enhanced Desktop Experience	Allowed	VDA for Server OS 7 through current VDA for Server OS

ICA/File Redirection

Name	Default setting	VDA
Auto connect client drives	Allowed	All VDA versions
Client drive redirection	Allowed	All VDA versions
Client fixed drives	Allowed	All VDA versions
Client floppy drives	Allowed	All VDA versions

Name	Default setting	VDA
Client network drives	Allowed	All VDA versions
Client optical drives	Allowed	All VDA versions
Client removable drives	Allowed	All VDA versions
Host to client redirection	Disabled	VDA for Server OS 7 through current VDA for Server OS
Preserve client drive letters	Disabled	VDA 5, 5.5, 5.6 Feature Pack 1, VDA for Desktop OS 7 through current VDA for Desktop OS
Read-only client drive access	Disabled	VDA 5.5, 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Special folder redirection	Allowed	Web Interface deployments only; VDA for Server OS 7 through current VDA for Server OS
Use asynchronous writes	Disabled	All VDA versions

ICA/Graphics

Name	Default setting	VDA
Display memory limit	65536 Kb	VDA 5, 5.5, 5.6 Feature Pack 1, VDA for Desktop OS 7 through current VDA for Desktop OS
Display mode degrade preference	Degrade color depth first	All VDA versions
Dynamic windows preview	Enabled	VDA 5.5, 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Image caching	Enabled	VDA 5.5, 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Legacy graphics mode	Disabled	VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Maximum allowed color depth	32 bits per pixel	All VDA versions
Notify user when	Disabled	VDA for Server OS 7 through current VDA for Server OS

display mode is degraded	Name	Default setting	VDA
Queuing and tossing	Enabled	All VDA versions	

ICA/Graphics/Caching

Name	Default setting	VDA
Persistent cache threshold	3000000 bps	VDA for Server OS 7 through current VDA for Server OS

ICA/Keep Alive

Name	Default setting	VDA
ICA keep alive timeout	60 seconds	All VDA versions
ICA keep alives	Do not send ICA keep alive messages	All VDA versions

ICA/Local App Access

Name	Default setting	VDA
Allow local app access	Prohibited	VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
URL redirection black list	No sites are specified	VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
URL redirection white list	No sites are specified	VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS

ICA/Mobile Experience

Name	Default setting	VDA
Automatic keyboard display	Prohibited	VDA 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Launch touch-optimized desktop	Allowed	VDA 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS This setting is disabled and not available for Windows 10 machines.

Remote the combo box	Prohibited	VDA 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS

ICA/Multimedia

Name	Default setting	VDA
Limit video quality	Not configured	VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Multimedia conferencing	Allowed	All VDA versions
Optimization for Windows Media multimedia redirection over WAN	Allowed	VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Use GPU for optimizing Windows Media multimedia redirection over WAN	Prohibited	VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Video load management policy setting	Not configured	VDA 7.6 FP3
Windows Media client-side content fetching	Allowed	VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Windows Media Redirection	Allowed	All VDA versions
Windows Media Redirection buffer size	5 seconds	VDA 5, 5.5, and 5.6, Feature Pack 1
Windows Media Redirection buffer size use	Disabled	VDA 5, 5.5, and 5.6, Feature Pack 1

ICA/Multi-Stream Connections

Name	Default setting	VDA
Audio over UDP	Allowed	VDA for Server OS 7 through current VDA for Server OS
Audio UDP port range	16500, 16509	VDA 5.5, 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Multi-Port policy	Primary port (2598) has High Priority	VDA 5.5, 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Multi-Stream computer setting	Disabled	VDA 5.5, 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Multi-Stream user setting	Disabled	VDA 5.5, 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS

ICA/Port Redirection

Name	Default setting	VDA
Auto connect client COM ports	Disabled	All VDA versions; for VDA 7.x, configure this setting using the registry.
Auto connect client LPT ports	Disabled	All VDA versions; for VDA 7.x, configure this setting using the registry.
Client COM port redirection	Prohibited	All VDA versions; for VDA 7.x, configure this setting using the registry.
Client LPT port redirection	Prohibited	All VDA versions; for VDA 7.x, configure this setting using the registry.

ICA/Printing

Name	Default setting	VDA
Client printer redirection	Allowed	All VDA versions
Default printer	Set default printer to the client's main printer	All VDA versions
Printer assignments	User's current printer is used as the default printer for the session	All VDA versions

Name	Default setting	VDA
Printer auto-creation event log preference	Log errors and warnings	All VDA versions
Session printers	No printers are specified	All VDA versions
Wait for printers to be created (desktop)	Disabled	All VDA versions

ICA/Printing/Client Printers

Name	Default setting	VDA
Auto-create client printers	Auto-create all client printers	All VDA versions
Auto-create generic universal printer	Disabled	All VDA versions
Client printer names	Standard printer names	All VDA versions
Direct connections to print servers	Enabled	All VDA versions
Printer driver mapping and compatibility	No rules are specified	All VDA versions
Printer properties retention	Held in profile only if not saved on client	All VDA versions
Retained and restored client printers	Allowed	VDA 5, 5.5 and 5.6 Feature Pack 1

ICA/Printing/Drivers

Name	Default setting	VDA
Automatic installation of in-box printer drivers	Enabled	All VDA versions
Universal driver preference	EMF; XPS; PCL5c; PCL4; PS	All VDA versions
Universal print driver usage	Use universal printing only if requested driver is unavailable	All VDA versions

ICA/Printing/Universal Print Server

Name	Default setting	VDA
Universal Print Server enable	Disabled	All VDA versions

Name	Default setting	VDA
Universal Print Server print data stream (CGP) port	7229	All VDA versions
Universal Print Server print stream input bandwidth limit (kpbs)	0	All VDA versions
Universal Print Server web service (HTTP/SOAP) port	8080	All VDA versions

ICA/Printing/Universal Printing

Name	Default setting	VDA
Universal printing EMF processing mode	Spool directly to printer	All VDA versions
Universal printing image compression limit	Best quality (lossless compression)	All VDA versions
Universal printing optimization defaults	Image Compression - Desired image quality = Standard quality - Enable heavyweight compression = False Image and Font Caching - Allow caching of embedded images = True - Allow caching of embedded fonts = True Allow non-administrators to modify these settings = False	All VDA versions
Universal printing preview preference	Do not use print preview for auto-created or generic universal printers	All VDA versions
Universal printing print quality limit	No limit	All VDA versions

ICA/Security

Name	Default setting	VDA
SecureICA minimum encryption level	Basic	VDA for Server OS 7 through current VDA for Server OS

ICA/Server Limits

Name	Default setting	VDA
Server idle timer interval	0 milliseconds	VDA for Server OS 7 through current VDA for Server OS

ICA/Session Limits

Name	Default setting	VDA
Disconnected session timer	Disabled	VDA 5, 5.5, 5.6 Feature Pack 1, VDA for Desktop OS 7 through current VDA for Desktop OS
Disconnected session timer interval	1440 minutes	VDA 5, 5.5, 5.6 Feature Pack 1, VDA for Desktop OS 7 through current VDA for Desktop OS
Session connection timer	Disabled	VDA 5, 5.5, 5.6 Feature Pack 1, VDA for Desktop OS 7 through current VDA for Desktop OS
Session connection timer interval	1440 minutes	VDA 5, 5.5, 5.6 Feature Pack 1, VDA for Desktop OS 7 through current VDA for Desktop OS
Session idle timer	Enabledf	VDA 5, 5.5, 5.6 Feature Pack 1, VDA for Desktop OS 7 through current VDA for Desktop OS
Session idle timer interval	1440 minutes	VDA 5, 5.5, 5.6 Feature Pack 1, VDA for Desktop OS 7 through current VDA for Desktop OS

ICA/Session Reliability

Name	Default setting	VDA
Session reliability connections	Allowed	All VDA versions
Session reliability port number	2598	All VDA versions
Session reliability timeout	180 seconds	All VDA versions

ICA/Time Zone Control

Name	Default setting	VDA
Estimate local time for legacy clients	Enabled	VDA for Server OS 7 through current VDA for Server OS
Use local time of client	Use server time zone	All VDA versions

ICA/TWAIN Devices

Name	Default setting	VDA
Client TWAIN device redirection	Allowed	VDA 5.5, 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
TWAIN compression level	Medium	VDA 5.5, 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS

ICA/USB Devices

Name	Default setting	VDA
Client USB device optimization rules	No rules are specified	VDA 7.6 FP3
Client USB device redirection	Prohibited	All VDA versions
Client USB device redirection rules	No rules are specified	All VDA versions
Client USB Plug and Play device redirection	Allowed	VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS

ICA/Visual Display

Name	Default setting	VDA
Preferred color depth for simple graphics	24 bits per pixel	VDA 7.6 FP3
Target frame rate	30 fps	All VDA versions
Visual quality	Medium	VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Use video codec for compression	Use video codec when available	VDA 7.6 FP3

ICA/Visual Display/Moving Images

Name	Default setting	VDA
Minimum image quality	Normal	VDA 5.5, 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Moving image compression	Enabled	VDA 5.5, 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Progressive compression level	None	VDA 5.5, 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Progressive compression threshold value	2147483647 Kbps	VDA 5.5, 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Target minimum frame rate	10 fps	VDA 5.5, 5.6 Feature Pack 1, VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS

ICA/Visual Display/Still Images

Name	Default setting	VDA
Extra color compression	Disabled	All VDA versions
Extra color compression threshold	8192 Kbps	All VDA versions
Heavyweight compression	Disabled	All VDA versions
Lossy compression level	Medium	All VDA versions
Lossy compression threshold value	2147483647 Kbps	All VDA versions

ICA/WebSockets

Name	Default setting	VDA
WebSockets connections	Prohibited	VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
WebSockets port number	8008	VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
WebSockets trusted origin server list	The wildcard, *, is used to trust all Receiver for Web URLs	VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS

Load Management

Name	Default setting	VDA
Concurrent logon tolerance	2	VDA for Server OS 7 through current VDA for Server OS
CPU usage	Disabled	VDA for Server OS 7 through current VDA for Server OS
CPU usage excluded process priority	Below Normal or Low	VDA for Server OS 7 through current VDA for Server OS
Disk usage	Disabled	VDA for Server OS 7 through current VDA for Server OS
Maximum number of sessions	250	VDA for Server OS 7 through current VDA for Server OS
Memory usage	Disabled	VDA for Server OS 7 through current VDA for Server OS
Memory usage base load	Zero load: 768MB	VDA for Server OS 7 through current VDA for Server OS

Profile Management/Advanced settings

Name	Default setting	VDA
Disable automatic configuration	Disabled	All VDA versions
Log off user if a problem is encountered	Disabled	All VDA versions
Number of retries when accessing locked files	5	All VDA versions
Process Internet cookie files on logoff	Disabled	All VDA versions

Profile Management/Basic settings

Name	Default setting	VDA
Active write back	Disabled	All VDA versions
Enable Profile management	Disabled	All VDA versions
Excluded groups	Disabled. Members of all user groups are processed.	All VDA versions
Offline profile support	Disabled	All VDA versions

Name	Default setting	VDA
Path to user store	Windows	All VDA versions
Process logons of local administrators	Disabled	All VDA versions
Processed groups	Disabled. Members of all user groups are processed.	All VDA versions

Profile Management/Cross-Platform Settings

Name	Default setting	VDA
Cross-platform settings user groups	Disabled. All user groups specified in Processed groups are processed	All VDA versions
Enable cross-platform settings	Disabled	All VDA versions
Path to cross-platform definitions	Disabled. No path is specified.	All VDA versions
Path to cross-platform settings store	Disabled. Windows\PM_CM is used.	All VDA versions
Source for creating cross-platform settings	Disabled	All VDA versions

Profile Management/File System/Exclusions

Name	Default setting	VDA
Exclusion list - directories	Disabled. All folders in the user profile are synchronized.	All VDA versions
Exclusion list - files	Disabled. All files in the user profile are synchronized.	All VDA versions

Profile Management/File System/Synchronization

Name	Default setting	VDA
Directories to synchronize	Disabled. Only non-excluded folders are synchronized.	All VDA versions
Files to synchronize	Disabled. Only non-excluded files are synchronized.	All VDA versions
Folders to mirror	Disabled. No folders are mirrored.	All VDA versions

Profile Management/Folder Redirection

Name	Default setting	VDA
Grant administrator access	Disabled	All VDA versions
Include domain name	Disabled	All VDA versions

Profile Management/Folder Redirection/AppData(Roaming)

Name	Default setting	VDA
AppData(Roaming) path	Disabled. No location is specified.	All VDA versions
Redirection settings for AppData(Roaming)	Contents are redirected to the UNC path specified in the AppData(Roaming) path policy settings	All VDA versions

Profile Management/Folder Redirection/Contacts

Name	Default setting	VDA
Contacts path	Disabled. No location is specified.	All VDA versions
Redirection settings for Contacts	Contents are redirected to the UNC path specified in the Contacts path policy settings	All VDA versions

Profile Management/Folder Redirection/Desktop

Name	Default setting	VDA
Desktop path	Disabled. No location is specified.	All VDA versions
Redirection settings for Desktop	Contents are redirected to the UNC path specified in the Desktop path policy settings	All VDA versions

Profile Management/Folder Redirection/Documents

Name	Default setting	VDA
Documents path	Disabled. No location is specified.	All VDA versions
Redirection settings for Documents	Contents are redirected to the UNC path specified in the Documents path policy settings	All VDA versions

Profile Management/Folder Redirection/Downloads

Name	Default setting	VDA
Downloads path	Disabled. No location is specified.	All VDA versions
Redirection settings for Downloads	Contents are redirected to the UNC path specified in the Downloads path policy settings	All VDA versions

Profile Management/Folder Redirection/Favorites

Name	Default setting	VDA
Favorites path	Disabled. No location is specified.	All VDA versions
Redirection settings for Favorites	Contents are redirected to the UNC path specified in the Favorites path policy settings	All VDA versions

Profile Management/Folder Redirection/Links

Name	Default setting	VDA
Links path	Disabled. No location is specified.	All VDA versions
Redirection settings for Links	Contents are redirected to the UNC path specified in the Links path policy settings	All VDA versions

Profile Management/Folder Redirection/Music

Name	Default setting	VDA
Music path	Disabled. No location is specified.	All VDA versions
Redirection settings for Music	Contents are redirected to the UNC path specified in the Music path policy settings	All VDA versions

Profile Management/Folder Redirection/Pictures

Name	Default setting	VDA
Pictures path	Disabled. No location is specified.	All VDA versions
Redirection settings for Pictures	Contents are redirected to the UNC path specified in the Pictures path policy settings	All VDA versions

Name	Default setting	VDA
Profile Management/Folder Redirection/Saved Games		
Name	Default setting	VDA
Saved Games path	Disabled. No location is specified.	All VDA versions
Redirection settings for Saved Games	Contents are redirected to the UNC path specified in the Saved Games path policy settings	All VDA versions

Profile Management/Folder Redirection/Searches

Name	Default setting	VDA
Searches path	Disabled. No location is specified.	All VDA versions
Redirection settings for Searches	Contents are redirected to the UNC path specified in the Searches path policy settings	All VDA versions

Profile Management/Folder Redirection/Start Menu

Name	Default setting	VDA
Start Menu path	Disabled. No location is specified.	All VDA versions
Redirection settings for Start Menu	Contents are redirected to the UNC path specified in the Start Menu path policy settings	All VDA versions

Profile Management/Folder Redirection/Video

Name	Default setting	VDA
Video path	Disabled. No location is specified.	All VDA versions
Redirection settings for Video	Contents are redirected to the UNC path specified in the Video path policy settings	All VDA versions

Profile Management/Log settings

Name	Default setting	VDA
Active Directory actions	Disabled	All VDA versions
Common information	Disabled	All VDA

Name	Default setting	versions VDA
Common warnings	Disabled	All VDA versions
Enable logging	Disabled	All VDA versions
File system actions	Disabled	All VDA versions
File system notifications	Disabled	All VDA versions
Logoff	Disabled	All VDA versions
Logon	Disabled	All VDA versions
Maximum size of the log file	1048576	All VDA versions
Path to log file	Disabled. Log files are saved in the default location; %SystemRoot%\System32\Logfiles\UserProfileManager.	All VDA versions
Personalized user information	Disabled	All VDA versions
Policy values at logon and logoff	Disabled	All VDA versions
Registry actions	Disabled	All VDA versions
Registry differences at logoff	Disabled	All VDA versions

Profile Management/Profile handling

Name	Default setting	VDA
Delay before deleting cached profiles	0	All VDA versions
Delete locally cached profiles on logoff	Disabled	All VDA versions
Local profile conflict handling	Use local profile	All VDA versions

Migration of existing profiles Name	Local and roaming Default setting	VDA versions
Path to the template profile	Disabled. New user profiles are created from the default user profile on the device where a user first logs on.	All VDA versions
Template profile overrides local profile	Disabled	All VDA versions
Template profile overrides roaming profile	Disabled	All VDA versions
Template profile used as a Citrix mandatory profile for all logons	Disabled	All VDA versions

Profile Management/Registry

Name	Default setting	VDA
Exclusion list	Disabled. All registry keys in the HKCU hive are processed when a user logs off.	All VDA versions
Inclusion list	Disabled. All registry keys in the HKCU hive are processed when a user logs off.	All VDA versions

Profile Management/Streamed user profiles

Name	Default setting	VDA
Always cache	Disabled	All VDA versions
Always cache size	0 Mb	All VDA versions
Profile streaming	Disabled	All VDA versions
Streamed user profile groups	Disabled. All user profiles within an OU are processed normally.	All VDA versions
Timeout for pending area lock files (days)	1 day	All VDA versions

Receiver

Name	Default setting	VDA
StoreFront	No stores are	VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server

accounts list Name	specified Default setting	OS and VDA for Desktop OS VDA

Virtual Delivery Agent

Name	Default setting	VDA
Controller registration IPv6 netmask	No netmask is specified	VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Controller registration port	80	All VDA versions
Controller SIDs	No SIDs are specified	All VDA versions
Controllers	No controllers are specified	All VDA versions
Enable auto update of controllers	Enabled	VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Only use IPv6 controller registration	Disabled	VDA for Server OS 7 and VDA for Desktop OS 7 through current VDA for Server OS and VDA for Desktop OS
Site GUID	No GUID is specified	All VDA versions

Virtual IP

Name	Default setting	VDA
Virtual IP loopback support	Disabled	VDA 7.6
Virtual IP virtual loopback programs list	None	VDA 7.6

HDX 3D Pro

Name	Default setting	VDA
Enable lossless	Enabled	VDA 5.5 and 5.6 Feature Pack 1
HDX 3D Pro quality settings		VDA 5.5 and 5.6 Feature Pack 1

策略设置参考

May 04, 2016

“策略”包含强制执行策略时应用的设置。本节中的说明还会指出要启用某项功能是否需要其他设置或与某项设置相似的其他设置。

快速引用

以下各表列举了可以在策略内配置的设置。请在左列中查找要完成的任务，然后在右列中找出相应的设置。

音频

对于此任务	使用此策略设置
控制是否允许使用多个音频设备	音频即插即用
控制是否允许从用户设备上的麦克风进行音频输入	客户端麦克风重定向
控制用户设备上的音频质量	音频质量
控制到用户设备上的扬声器的音频映射	客户端音频重定向

用户设备带宽

要限制用于以下项目的带宽	使用此策略设置
客户端音频映射	• 音频重定向带宽限制或 • 音频重定向带宽限制百分比
使用本地剪贴板执行的剪切和粘贴操作	• 剪贴板重定向带宽限制或 • 剪贴板重定向带宽限制百分比
会话中对本地客户端驱动器的访问	• 文件重定向带宽限制或 • 文件重定向带宽限制百分比
HDX MediaStream 多媒体加速	• HDX MediaStream 多媒体加速带宽限制或 • HDX MediaStream 多媒体加速带宽限制百分比
客户端会话	总会话带宽限制

打印机 要限制用于以下项目的带宽	• 打印机重定向带宽限制或 • 使用此策略设置 • 打印机重定向带宽限制百分比
TWAIN 设备（例如照相机或扫描仪）	• TWAIN 设备重定向带宽限制或 • TWAIN 设备重定向带宽限制百分比
USB 设备	• 客户端 USB 设备重定向带宽限制或 • 客户端 USB 设备重定向带宽限制百分比

客户端驱动器和用户设备的重定向

对于此任务	使用此策略设置
控制是否在用户登录到服务器时连接用户设备上的驱动器	自动连接客户端驱动器
控制服务器与本地剪贴板之间的剪切-粘贴式数据传输	客户端剪贴板重定向
控制从用户设备映射驱动器的方式	客户端驱动器重定向
控制在会话中可否使用用户的本地硬盘驱动器	• 客户端固定驱动器和 • 客户端驱动器重定向
控制在会话中可否使用用户的本地软盘驱动器	• 客户端软盘驱动器和 • 客户端驱动器重定向
控制在会话中可否使用用户的网络驱动器	• 客户端网络驱动器和 • 客户端驱动器重定向
控制在会话中可否使用用户的本地 CD、DVD、或蓝光驱动器	• 客户端光盘驱动器和 • 客户端驱动器重定向
控制在会话中可否使用用户的本地可移动驱动器	• 客户端可移动驱动器和 • 客户端驱动器重定向
控制在会话中可否使用用户的 TWAIN 设备（如扫描仪和相机），并控制图像数据传输的压缩	• 客户端 TWAIN 设备重定向 • TWAIN 压缩重定向
控制在会话中可否使用 USB 设备	• 客户端 USB 设备重定向和 • 客户端 USB 设备重定向规则

对于此任务 通过 WAN 将文件写入和复制到客户端磁盘的速度	使用此策略设置

内容重定向

对于此任务	使用此策略设置
控制是否要使用从服务器到用户设备的内容重定向	主机到客户端重定向

桌面 UI

对于此任务	使用此策略设置
控制是否在用户会话中使用桌面墙纸	桌面墙纸
拖动窗口时查看窗口内容	拖动时查看窗口内容

图形和多媒体

对于此任务	使用此策略设置
控制每秒从虚拟桌面发送到用户设备的最大帧数	目标帧速率
控制用户设备上显示的图像的视觉质量	视觉质量
控制是否在会话中呈现 Flash 内容	Flash 默认行为
控制在会话中访问时 Web 站点可否显示 Flash 内容	- Flash 服务器端内容提取 URL 列表 - Flash URL 兼容性列表 - Flash 视频回退预防策略设置 - Flash 视频回退预防错误 *.swf

确定多流网络流量优先级

对于此任务	使用此策略设置
为跨多个连接的 ICA 通信指定端口并确定网络优先级	多端口策略

启用对服务器与用户设备之间多流连接的支持 对于此任务	多流 (计算机和用户设置) 使用此策略设置

打印

对于此任务	使用此策略设置
控制在用户设备上创建客户端打印机的行为	- 自动创建客户端打印机和 - 客户端打印机重定向
控制打印机属性的存储位置	打印机属性保留
控制由客户端还是服务器处理打印请求	直接连接到打印服务器
控制用户可否访问连接到其用户设备的打印机	客户端打印机重定向
控制自动创建客户端和网络打印机时的本机 Windows 驱动程序的安装	自动安装现成的打印机驱动程序
控制何时使用通用打印机驱动程序	通用打印驱动程序使用
根据漫游用户的会话信息选择打印机	默认打印机

注意：在桌面或应用程序会话中不能使用策略来启用屏幕保护程序。如果用户需要启用屏幕保护程序，可以在用户设备上实现。

ICA 策略设置

May 04, 2016

ICA 部分包含与 ICA 侦听器连接和到剪贴板的映射相关的策略设置。

客户端剪贴板重定向

此设置允许或阻止将用户设备上的剪贴板映射到服务器上的剪贴板。

默认情况下，允许剪贴板重定向。

要阻止剪贴数据在会话与本地剪贴板之间传输，请选择禁止。用户仍可以在会话中运行的应用程序之间剪切和粘贴数据。

启用此设置之后，使用剪贴板重定向带宽限制或剪贴板重定向带宽限制百分比设置来配置剪贴板在客户端连接中可以占用的最大允许带宽。

客户端剪贴板写入允许的格式

启用限制客户端剪贴板写入设置时，主机剪贴板数据无法与客户端端点共享，但是您可以利用此设置来允许与客户端端点剪贴板共享指定的数据格式。要使用此设置，请启用此设置并添加允许的指定格式。

以下剪贴板格式是系统定义的格式：

- CF_TEXT
- CF_BITMAP
- CF_METAFILEPICT
- CF_SYLK
- CF_DIF
- CF_TIFF
- CF_OEMTEXT
- CF_DIB
- CF_PALETTE
- CF_PENDATA
- CF_RIFF
- CF_WAVE
- CF_UNICODETEXT
- CF_ENHMETAFILE
- CF_HDROP
- CF_LOCALE
- CF_DIBV5
- CF_OWNERDISPLAY
- CF_DSPTEXT
- CF_DSPBITMAP
- CF_DSPMETAFILEPICT
- CF_DISPENHMETAFILE

以下自定义格式是 XenApp 和 XenDesktop 中的预定义格式：

- CFX_RICHTEXT
- CFX_OfficeDrawingShape
- CFX_BIFF8

可以添加其他自定义格式。自定义格式名称必须与要向系统注册的格式匹配。格式名称区分大小写。

如果将客户端剪贴板重定向或限制客户端剪贴板写入设置为“禁止”，将无法应用此设置。

桌面启动

此设置允许或阻止 VDA 直接访问用户组中的非管理员用户使用 ICA 连接来连接到该 VDA 上的会话。

默认情况下，非管理用户无法连接到这些会话。

此设置对 VDA 直接访问用户组中使用 RDP 连接的非管理员用户没有影响；不管是否启用此设置，这些用户都可以连接到 VDA。此设置对不属于 VDA 直接访问用户组的非管理员用户没有影响；不管是否启用此设置，这些用户都不可以连接到 VDA。

ICA 侦听器连接超时

注意：此设置仅适用于以下 Virtual Delivery Agent：5.0、5.5 和 5.6 Feature Pack 1。

此设置指定完成使用 ICA 协议的连接需要等待的最长时间。

默认情况下，需要等待的最长时间为 120000 毫秒（或两分钟）。

ICA 侦听器端口号

此设置指定服务器上 ICA 协议使用的 TCP/IP 端口号。

默认情况下，此端口号设置为 1494。

有效端口号必须在介于 0 到 65535 的范围内，且不得与其他已知端口号相冲突。如果更改端口号，请重新启动服务器，新值才能生效。如果在服务器上更改端口号，还必须在每个连接到该服务器的 Receiver 或插件上更改该端口号。

在客户端连接期间启动非发布程序

此设置指定是否允许通过服务器上的 RDP 启动初始应用程序。

默认情况下，不允许通过服务器上的 RDP 启动初始应用程序。

限制客户端剪贴板写入

如果将其设置为“允许”，主机剪贴板数据无法与客户端端点共享。可以通过启用客户端剪贴板写入允许的格式设置允许特定格式。

默认情况下，此设置为“禁止”。

限制会话剪贴板写入

此设置为“允许”时，客户端剪贴板数据无法在用户会话中共享。可以通过启用会话剪贴板写入允许的格式设置允许特定格式。

默认情况下，此设置为“禁止”。

会话剪贴板写入允许的格式

将限制会话剪贴板写入设置为“允许”时，客户端剪贴板数据无法与会话应用程序共享，但是您可以利用此设置来允许与会话剪贴板共享指定的数据格式。

以下剪贴板格式是系统定义的格式：

- CF_TEXT
- CF_BITMAP
- CF_METAFILEPICT
- CF_SYLK
- CF_DIF
- CF_TIFF
- CF_OEMTEXT
- CF_DIB
- CF_PALETTE
- CF_PENDATA
- CF_RIFF
- CF_WAVE
- CF_UNICODETEXT
- CF_ENHMETAFILE
- CF_HDROP
- CF_LOCALE
- CF_DIBV5
- CF_OWNERDISPLAY
- CF_DSPTEXT
- CF_DSPBITMAP
- CF_DSPMETAFILEPICT
- CF_DISPENHMETAFILE

以下自定义格式是 XenApp 和 XenDesktop 中的预定义格式：

- CFX_RICHTEXT
- CFX_OfficeDrawingShape
- CFX_BIFF8

可以添加其他自定义格式。自定义格式名称必须与要向系统注册的格式匹配。格式名称区分大小写。

如果将客户端剪贴板重定向或限制会话剪贴板写入设置为“禁止”，将无法应用此设置。

客户端自动重新连接策略设置

May 04, 2016

“客户端重新连接”部分包含用于控制会话自动重新连接的策略设置。

客户端自动重新连接

此设置允许或阻止同一客户端在连接中断后自动重新连接。

默认情况下，允许自动重新连接。

如果允许自动重新连接，则当连接断开时，用户将可以在从中断处继续执行原来的工作。自动重新连接会检测连接断开情形，然后将用户重新连接到其会话。

但是，如果未使用 Receiver 的 Cookie（包括会话 ID 的密钥和凭据），自动重新连接可能会导致启动新的会话（而不会重新连接到现有会话）。如果 Cookie 已过期（例如因为重新连接延迟，或者必须重新输入凭据），则不使用该 Cookie。如果用户有意断开连接，则不触发客户端自动重新连接。

对于应用程序会话，允许自动重新连接时，Receiver 将一直尝试重新连接会话，直到重新连接成功或者用户取消重新连接尝试。

对于桌面会话，允许自动重新连接时，Receiver 将在指定时间段内尝试重新连接会话，除非重新连接成功或者用户取消了重新连接尝试。默认情况下，此时间段为五分钟。要更改此时间段，请在用户设备上编辑以下注册表项：

HKLM\Software\Citrix\ICA Client\TransportReconnectRetryMaxTimeSeconds; DWORD;<秒数>

其中，<秒数> 是以秒为单位的数字，经过这些秒数后，将不再尝试重新连接会话。

客户端自动重新连接身份验证

此设置要求对客户端自动重新连接进行身份验证。

默认情况下，无需进行身份验证。

在用户最初登录时，其凭据将加密并存储在内存中，并创建一个包含发送到 Receiver 的加密密钥的 Cookie。配置此设置后，将不使用 Cookie。而是在 Receiver 试图自动重新连接时，向用户显示一个对话框，要求输入凭据。

客户端自动重新连接日志记录

此设置启用或禁用将事件日志中记录客户端自动重新连接。

默认情况下，禁用日志记录。

启用日志记录后，服务器的系统日志将捕获与成功或失败的自动重新连接事件有关的信息。站点并不会提供所有服务器的重新连接事件组合日志。

音频策略设置

May 04, 2016

“音频”部分包含的策略设置允许用户设备在会话中发送和接收音频，而不会降低性能。

通过 UDP 实时传输音频

此设置可允许或阻止使用用户数据报协议 (User Datagram Protocol, UDP) 通过 RTP 在 VDA 和用户设备之间传输和接收音频的功能。禁用此设置后，将通过 TCP 发送和接收音频。

默认情况下，允许通过 UDP 传输音频。

音频即插即用

该设置允许或阻止适用多个音频设备来记录和播放声音。

默认情况下，允许使用多个音频设备。

此设置仅适用于 Windows 服务器操作系统计算机。

音频质量

该设置指定用户会话所接收的声音的质量等级。

默认情况下，音频质量设置为高 - 高清晰度音频。

要控制音频质量，请选择以下选项之一：

- 对于低带宽连接，请选择低 - 适用于低速连接。发送给用户设备的音频最高可压缩为 16 Kbps。这种压缩会大幅降低低带宽连接的音频质量，但是可以获得合理的性能。
- 选择中 - 语音优化可交付 IP 语音 (VoIP) 应用程序，或者在使用低于 512 Kbps 的线路或发生严重网络拥堵和数据包丢失的网络连接中交付媒体应用程序。此编解码器能够快速编码，非常适合在需要服务器端媒体处理时与软件电话和统一通信应用程序结合使用。
发送给用户设备的音频最多可压缩至 64 Kbps。此压缩级别会导致用户设备上播放的音频质量适当下降，但是可缩短延迟并仅占用很少的带宽。如果 VoIP 质量无法满足需要，请确保将通过 UDP 协议的音频实时传输策略设置为允许。
- 对于带宽足够且音频质量很重要的连接，请选择高 - 高清晰度音频。客户端可以按照其本机速率播放声音。声音在保持最高达 CD 质量的高质量级别压缩，并使用最高 112 Kbps 的带宽。传输此数据量可能会导致 CPU 使用率增加以及网络拥塞。

只有在录制或播放音频时，才会占用带宽。如果两者同时发生，则会占用双倍带宽。

要指定最大带宽量，请配置音频重定向带宽限制或音频重定向带宽限制百分比设置。

客户端音频重定向

此设置指定托管在服务器上的应用程序是否可以通过安装在用户设备上的音频设备来播放声音。此设置还指定用户是否可以录制音频输入。

默认情况下，允许音频重定向。

允许此设置之后，可以限制播放或录制音频占用的带宽。限制音频占用的带宽量可提高应用程序性能，但也可能会降低音频质量。只有在录制或播放音频时，才会占用带宽。如果两者同时发生，则会占用双倍带宽。要指定最大带宽量，请配置音频重定向带宽限制或音频重定向带宽限制百分比设置。

在 Windows 服务器操作系统计算机上，请确保将音频即插即用设置为启用以支持多个音频设备。

重要：禁止客户端音频重定向将禁用所有 HDX 音频功能。

客户端麦克风重定向

此设置启用或禁用客户端麦克风重定向。启用后，用户在会话中可以使用麦克风录制音频输入。

默认情况下，允许麦克风重定向。

出于安全考虑，当不受用户设备信任的服务器尝试访问麦克风时，系统会向用户发出警报。用户可以选择是否接受访问。用户可以在 Citrix Receiver 上禁用警报。

在 Windows 服务器操作系统计算机上，请确保将音频即插即用设置为启用以支持多个音频设备。

如果在用户设备上禁用了客户端音频重定向设置，则此规则不起任何作用。

带宽策略设置

May 04, 2016

“带宽”部分包含的一些策略设置可避免出现与客户端会话带宽使用有关的性能问题。

重要：将这些策略设置与多流策略设置结合使用时，可能会导致意外的结果。如果在某个策略中使用“多流”设置，请确保不要在该策略中包含这些带宽限制策略设置。

音频重定向带宽限制

此设置指定在用户会话中播放或录制音频时允许使用的最大带宽 (Kbps)。

默认情况下，未指定最大值（零）。

如果为此设置输入一个值，并为音频重定向带宽限制百分比设置输入一个值，则应用最严格（具有较低值）的设置。

音频重定向带宽限制百分比

此设置指定播放或录制音频时允许使用的最大带宽（占总会话带宽的百分比）。

默认情况下，未指定最大值（零）。

如果为此设置输入一个值，并为音频重定向带宽限制设置输入一个值，则应用最严格（具有较低值）的设置。

如果配置此设置，还必须配置总会话带宽限制设置，后者用于指定客户端会话可用的带宽总量

客户端 USB 设备重定向带宽限制

此设置指定允许往来于客户端的 USB 设备重定向所使用的最大带宽 (Kbps)。

默认情况下，未指定最大值（零）。

如果针对此设置和客户端 USB 设备重定向带宽限制百分比设置都输入了值，将应用最严格（具有较低值）的设置。

客户端 USB 设备重定向带宽限制百分比

此设置指定允许往来于客户端的 USB 设备重定向所使用的最大带宽（占总会话带宽的百分比）。

默认情况下，未指定最大值（零）。

如果针对此设置和客户端 USB 设备重定向带宽限制设置都输入了值，将应用最严格（具有较低值）的设置。

如果配置此设置，还必须配置总会话带宽限制设置，后者用于指定客户端会话可用的带宽总量

剪贴板重定向带宽限制

此设置指定在会话和本地剪贴板之间传输数据时允许使用的最大带宽 (Kbps)。

默认情况下，未指定最大值（零）。

如果为此设置输入一个值，并为剪贴板重定向带宽限制百分比设置输入一个值，则应用最严格（具有较低值）的设置。

剪贴板重定向带宽限制百分比

此设置指定在会话和本地剪贴板之间传输数据时允许使用的最大带宽（占总会话带宽的百分比）。

默认情况下，未指定最大值（零）。

如果为此设置输入一个值，并为剪贴板重定向带宽限制设置输入一个值，则应用最严格（具有较低值）的设置。

如果配置此设置，还必须配置总会话带宽限制设置，后者用于指定客户端会话可用的带宽总量

COM 端口重定向带宽限制

注意：对于 Virtual Delivery Agent 7.x，请使用注册表配置此设置。请参阅[使用注册表配置 COM 端口和 LPT 端口重定向设置](#)。

此设置指定在客户端连接中访问 COM 端口时允许使用的最大带宽 (Kbps)。如果为此设置输入一个值，并为 COM 端口重定向带宽限制百分比设置输入一个值，则应用最严格（具有较低值）的设置。

COM 端口重定向带宽限制百分比

注意：对于 Virtual Delivery Agent 7.x，请使用注册表配置此设置。请参阅[使用注册表配置 COM 端口和 LPT 端口重定向设置](#)。

此设置指定在客户端连接中访问 COM 端口时允许使用的最大带宽（占总会话带宽的百分比）。

默认情况下，未指定最大值（零）

如果为此设置输入一个值，并为 COM 端口重定向带宽限制设置输入一个值，则应用最严格（具有较低值）的设置。

如果配置此设置，还必须配置总会话带宽限制设置，后者用于指定客户端会话可用的带宽总量

文件重定向带宽限制

此设置指定在用户会话中访问客户端驱动器时允许使用的最大带宽 (Kbps)。

默认情况下，未指定最大值（零）。

如果为此设置输入一个值，并为文件重定向带宽限制百分比设置输入一个值，则应用最严格（具有较低值）的设置。

文件重定向带宽限制百分比

此设置指定访问客户端驱动器时允许使用的最大带宽（占总会话带宽的百分比）。

默认情况下，未指定最大值（零）。

如果为此设置输入一个值，并为文件重定向带宽限制设置输入一个值，则应用最严格（具有较低值）的设置。

如果配置此设置，还必须配置总会话带宽限制设置，后者用于指定客户端会话可用的带宽总量

HDX MediaStream 多媒体加速带宽限制

此设置指定在通过 HDX MediaStream 多媒体加速交付流音频和视频时允许使用的最大带宽 (Kbps)。

默认情况下，未指定最大值（零）。

如果为此设置以及 HDX MediaStream 多媒体加速带宽限制百分比设置都输入值，将应用最严格的设置（两者中较小者）。

HDX MediaStream 多媒体加速带宽限制百分比

此设置指定在通过 HDX MediaStream 多媒体加速交付流音频和视频时允许使用的最大带宽（占总会话带宽的百分比）。

默认情况下，未指定最大值（零）。

如果为此设置以及 HDX MediaStream 多媒体加速带宽限制设置都输入值，则将应用最严格的设置（两者中较小者）。

如果配置此设置，还必须配置总会话带宽限制设置，后者用于指定客户端会话可用的带宽总量

LPT 端口重定向带宽限制

注意：对于 Virtual Delivery Agent 7.x，请使用注册表配置此设置。请参阅[使用注册表配置 COM 端口和 LPT 端口重定向设置](#)。

此设置指定在单个用户会话中使用 LPT 端口的打印作业允许使用的最大带宽 (Kbps)。

默认情况下，未指定最大值（零）。

如果为此设置输入一个值，并为 LPT 端口重定向带宽限制百分比设置输入一个值，则应用最严格（具有较低值）的设置。

LPT 端口重定向带宽限制百分比

注意：对于 Virtual Delivery Agent 7.x，请使用注册表配置此设置。请参阅[使用注册表配置 COM 端口和 LPT 端口重定向设置](#)。

此设置指定在单个客户端会话中使用 LPT 端口的打印作业的带宽限制（占总会话带宽的百分比）。

默认情况下，未指定最大值（零）。

如果为此设置输入一个值，并为 LPT 端口重定向带宽限制设置输入一个值，则应用最严格（具有较低值）的设置。

如果配置此设置，还必须配置总会话带宽限制设置，后者用于指定客户端会话可用的带宽总量

总会话带宽限制

此设置指定用户会话可用的总带宽 (Kbps)。

最大可强制带宽上线为 10 Mbps (10,000 Kbps)。默认情况下，未指定最大值（零）。

当客户端连接外的其他应用程序竞用有限带宽时，限制客户端连接所占用的带宽量可提高性能。

打印机重定向带宽限制

此设置指定在用户会话中访问客户端打印机时允许使用的最大带宽 (Kbps)。

默认情况下，未指定最大值（零）。

如果为此设置输入一个值，并为打印机重定向带宽限制百分比设置输入一个值，则应用最严格（具有较低值）的设置。

打印机重定向带宽限制百分比

此设置指定访问客户端打印机时允许使用的最大带宽（占总会话带宽的百分比）。

默认情况下，未指定最大值（零）。

如果为此设置输入一个值，并为打印机重定向带宽限制设置输入一个值，则应用最严格（具有较低值）的设置。

如果配置此设置，还必须配置总会话带宽限制设置，后者用于指定客户端会话可用的带宽总量

TWAIN 设备重定向带宽限制

此设置指定从已发布的应用程序控制 TWAIN 成像设备时允许使用的最大带宽 (Kbps)。

默认情况下，未指定最大值（零）。

如果为此设置输入一个值，并为 TWAIN 设备重定向带宽限制百分比设置输入一个值，则应用最严格（具有较低值）的设置。

TWAIN 设备重定向带宽限制百分比

此设置指定从已发布的应用程序控制 TWAIN 成像设备时允许使用的最大带宽（占总会话带宽的百分比）。

默认情况下，未指定最大值（零）。

如果为此设置输入一个值，并为 TWAIN 设备重定向带宽限制设置输入一个值，则应用最严格（具有较低值）的设置。

如果配置此设置，还必须配置总会话带宽限制设置，后者用于指定客户端会话可用的带宽总量

客户端传感器策略设置

May 04, 2016

“客户端传感器”部分中包含用于控制如何在用户会话中处理移动设备传感器信息的策略设置。

允许应用程序使用客户端设备的物理位置

此设置决定是否允许在移动设备上的会话中运行的应用程序使用用户设备的物理位置。

默认情况下，禁止使用位置信息。

如果禁用了此设置，则应用程序尝试检索位置信息时将返回值“权限遭拒”。

如果启用了此设置，用户可以通过拒绝 Receiver 访问位置的请求来禁止使用位置信息。Receiver 首次发出请求时，Android 和 iOS 设备将提示在每个会话中输入位置信息。

开发使用允许应用程序使用客户端设备的物理位置设置的托管应用程序时，请注意以下各项：

- 启用了位置功能的应用程序不应依赖于当前可用的位置信息，原因如下：
 - 用户可能不允许访问位置信息。
 - 位置可能不可用，或者在应用程序运行过程中可能会发生变化。
 - 用户可能会从不支持位置信息的其他设备连接到应用程序会话。
- 启用了位置功能的应用程序必须满足以下条件：
 - 默认关闭位置功能。
 - 提供一个用户选项，用于在应用程序运行过程中启用或禁用位置功能。
 - 提供一个用户选项，用于清除应用程序缓存的位置数据。（Receiver 不缓存位置数据。）
- 启用了位置功能的应用程序必须精确管理位置信息，以便获取的数据能够满足应用程序的需求，且遵守所有相关司法管辖区的法规。
- 使用定位服务时，应强制使用安全连接（例如，使用 SSL/TLS 或 VPN）。Citrix Receiver 应连接到可信服务器。
- 注意征求使用定位服务方面的法律意见。

桌面 UI 策略设置

Nov 02, 2016

The Desktop UI section contains policy settings that control visual effects such as desktop wallpaper, menu animations, and drag-and-drop images, to manage the bandwidth used in client connections. You can improve application performance on a WAN by limiting bandwidth usage.

Desktop Composition Redirection

This setting specifies whether to use the processing capabilities of the graphics processing unit (GPU) or integrated graphics processor (IGP) on the user device for local DirectX graphics rendering to provide users with a more fluid Windows desktop experience. When enabled, Desktop Composition Redirection delivers a highly responsive Windows experience while maintaining high scalability on the server.

By default, Desktop Composition Redirection is disabled (7.6 FP3 through current) and enabled (5.6 through 7.6 FP2).

To turn off Desktop Composition Redirection and reduce the bandwidth required in user sessions, select Disabled when adding this setting to a policy.

Desktop Composition Redirection graphics quality

This setting specifies the quality of graphics used for Desktop Composition Redirection.

By default, this is set to high.

Choose from High, Medium, Low, or Lossless quality.

Desktop wallpaper

This setting allows or prevents wallpaper showing in user sessions.

By default, user sessions can show wallpaper.

To turn off desktop wallpaper and reduce the bandwidth required in user sessions, select Prohibited when adding this setting to a policy.

Menu animation

This setting allows or prevents menu animation in user sessions.

By default, menu animation is allowed.

Menu animation is a Microsoft personal preference setting for ease of access. When enabled, it causes a menu to appear after a short delay, either by scrolling or fading in. An arrow icon appears at the bottom of the menu. The menu appears when you point to that arrow.

Menu animation is enabled on a desktop if this policy setting is set to Allowed and the menu animation Microsoft personal preference setting is enabled.

Note: Changes to the menu animation Microsoft personal preference setting are changes to the desktop. This means that if the desktop is set to discard changes when the session ends, a user who has enabled menu animations in a session may not have menu animation available in subsequent sessions on the desktop. For users who require menu animation, enable

the Microsoft setting in the master image for the desktop or ensure that the desktop retains user changes.

View window contents while dragging

This setting allows or prevents the display of window contents when dragging a window across the screen.

By default, viewing window contents is allowed.

When set to Allowed, the entire window appears to move when you drag it. When set to Prohibited, only the window outline appears to move until you drop it.

最终用户监控策略设置

May 04, 2016

“最终用户监控”部分包含用于测量会话流量的策略设置。

ICA 往返程计算

此设置确定是否为活动连接执行 ICA 往返程计算。

默认情况下，启用活动连接的计算。

默认情况下，每次 ICA 往返程度量的启动都将延迟，直至指示有用户交互的通信流出现。此延迟的长度不限，以防止 ICA 往返程度量成为产生 ICA 通信流的唯一原因。

ICA 往返程计算间隔

此设置指定 ICA 往返程计算的执行间隔（以秒为单位）。

默认情况下，ICA 往返程每 15 秒钟计算一次。

空闲连接的 ICA 往返程计算

此设置确定是否为空闲连接执行 ICA 往返程计算。

默认情况下，不为空闲连接执行计算。

默认情况下，每次 ICA 往返程度量的启动都将延迟，直至指示有用户交互的通信流出现。此延迟的长度不限，以防止 ICA 往返程度量成为产生 ICA 通信流的唯一原因。

Enhanced Desktop Experience 策略设置

May 04, 2016

Enhanced Desktop Experience 策略设置可配置在服务器操作系统上运行的会话，使其看起来像本地 Windows 7 桌面，从而为用户提供增强的桌面体验。

默认情况下，启用此设置。

如果虚拟桌面上已存在具有 Windows Classic 主题的用户配置文件，则启用此策略将不会为该用户提供增强的桌面体验。如果用户以 Windows 7 主题用户配置文件登录到运行未配置或禁用此策略的 Windows Server 2012 的虚拟桌面，则会向该用户显示错误消息，指明应用主题失败。

在上述两种情况下，重置用户配置文件即可解决问题。

如果策略在具有活动用户会话的虚拟桌面上从启用状态更改为禁用状态，则这些会话的外观会与 Windows 7 和 Windows Classic 桌面体验不一致。为避免出现此问题，请确保在更改此策略设置后重新启动虚拟桌面。您还必须删除虚拟桌面上的任何漫游配置文件。Citrix 还建议您删除虚拟桌面上的任何其他用户配置文件，以避免配置文件之间的不一致。

如果您在环境中使用漫游用户配置文件，请确保对共享同一配置文件的所有虚拟桌面启用或禁用 Enhanced Desktop Experience 功能。

Citrix 建议不要在运行服务器操作系统和客户端操作系统的虚拟桌面之间共享漫游配置文件。适用于客户端和服务器的配置文件的配置有所差别，跨两种类型的操作系统共享漫游配置文件可能会导致用户在这两种操作系统之间移动时配置文件属性不一致。

文件重定向策略设置

May 04, 2016

“文件重定向”部分包含与客户端驱动器映射和客户端驱动器优化有关的策略设置。

自动连接客户端驱动器

此设置允许或禁止在用户登录时自动连接客户端驱动器。

默认情况下允许自动连接。

将此设置添加到策略时，请确保启用您希望自动连接的驱动器类型的设置。例如，要允许自动连接到用户的 CD-ROM 驱动器，请配置此设置以及客户端光盘驱动器设置。

下列策略设置为相关设置：

- 客户端驱动器重定向
- 客户端软盘驱动器
- 客户端光盘驱动器
- 客户端固定驱动器
- 客户端网络驱动器
- 客户端可移动驱动器

客户端驱动器重定向

此设置启用或禁用往来于用户设备上的驱动器的文件重定向。

默认情况下，启用文件重定向。

启用时，用户可将文件保存到你所有客户端驱动器。禁用时，将禁止所有文件重定向，而不考虑各文件重定向设置（例如客户端软盘驱动器和客户端网络驱动器）的状态。

下列策略设置为相关设置：

- 客户端软盘驱动器
- 客户端光盘驱动器
- 客户端固定驱动器
- 客户端网络驱动器
- 客户端可移动驱动器

客户端固定驱动器

此设置允许或禁止用户访问文件或将文件保存到用户设备上的固定驱动器。

默认情况下，允许访问客户端固定驱动器。

将此设置添加到策略中时，请确保客户端驱动器重定向设置存在并且设置为允许。如果禁用这些设置，将不映射客户端固定驱动器，且用户无法手动访问这些驱动器，无论客户端固定驱动器设置的状态如何。

要确保在用户登录时自动连接固定驱动器，请配置自动连接客户端驱动器设置。

客户端软盘驱动器

此设置允许或禁止用户访问文件或将文件保存到用户设备上的软盘驱动器。

默认情况下，允许访问客户端软盘驱动器。

将此设置添加到策略中时，请确保客户端驱动器重定向设置存在并且设置为允许。如果禁用这些设置，将不映射客户端固定驱动器，且用户无法手动访问这些驱动器，无论客户端软盘驱动器设置的状态如何。

要确保在用户登录时自动连接软盘驱动器，请配置自动连接客户端驱动器设置。

客户端网络驱动器

此设置允许或禁止用户通过用户设备访问文件或将文件保存到网络（远程）驱动器。

默认情况下，允许访问客户端网络驱动器。

将此设置添加到策略中时，请确保客户端驱动器重定向设置存在并且设置为允许。如果禁用这些设置，将不映射客户端网络驱动器，且用户无法手动访问这些驱动器，无论客户端网络驱动器设置的状态如何。

要确保在用户登录时自动连接网络驱动器，请配置自动连接客户端驱动器设置。

客户端光盘驱动器

此设置允许或禁止用户访问文件或将文件保存到用户设备上的 CD-ROM、DVD-ROM 和 BD-ROM 驱动器。

默认情况下，允许访问客户端光盘驱动器。

将此设置添加到策略中时，请确保客户端驱动器重定向设置存在并且设置为允许。如果禁用这些设置，将不映射客户端光盘驱动器，且用户无法手动访问这些驱动器，无论客户端光盘驱动器设置的状态如何。

要确保在用户登录时自动连接光盘驱动器，请配置自动连接客户端驱动器设置。

客户端可移动驱动器

此设置允许或禁止用户访问文件或将文件保存到用户设备上的 USB 驱动器。

默认情况下，允许访问客户端可移动驱动器。

将此设置添加到策略中时，请确保客户端驱动器重定向设置存在并且设置为允许。如果禁用这些设置，将不映射客户端可移动驱动器，且用户无法手动访问这些驱动器，无论客户端可移动驱动器设置的状态如何。

要确保在用户登录时自动连接可移动驱动器，请配置自动连接客户端驱动器设置。

主机到客户端重定向

此设置启用或禁用将在用户设备上打开的 URL 及某些媒体内容的文件类型关联。禁用时，内容在服务器上打开。

默认情况下，禁用文件类型关联。

启用此设置后，以下这些 URL 类型将在本地打开：

- 超文本传输协议 (HTTP)
- 安全超文本传输协议 (HTTPS)
- Real Player 和 QuickTime (RTSP)
- Real Player 和 QuickTime (RTSPU)
- 旧版 Real Player (PNM)
- Microsoft Media Server (MMS)

保留客户端驱动器盘符

此设置允许或禁止将客户端驱动器映射到会话中的同一驱动器号。

默认情况下，不保留客户端驱动器盘符。

将此设置添加到策略中时，请确保客户端驱动器重定向设置存在并且设置为允许。

只读客户端驱动器访问

该设置允许或阻止用户及应用程序创建或修改映射的客户端驱动器上的文件或文件夹。

默认情况下，可以修改映射的客户端驱动器上的文件和文件夹。

如果设置为已启用，则具有只读权限即可访问文件和文件夹。

将此设置添加到策略中时，请确保客户端驱动器重定向设置存在并且设置为允许。

特殊文件夹重定向

此设置允许或阻止 Citrix Receiver 和 Web Interface 用户从会话中看到其本地文档和桌面特殊文件夹。

默认情况下，允许特殊文件夹重定向。

此设置可防止任何通过策略过滤的对象使用特殊文件夹重定向，无论其他位置存在何种设置。如果禁止此设置，将忽略为 StoreFront、Web Interface 或 Citrix Receiver 指定的任何相关设置。

要定义哪些用户可以使用特殊文件夹重定向，请选择允许，并将此设置包括在对您希望具有此功能的用户执行过滤的策略中。此设置将覆盖所有其他特殊文件夹重定向设置。

由于特殊文件夹重定向必须与用户设备交互，因此，禁止用户访问文件或将文件保存到本地硬盘驱动器的策略设置，也会禁止用户使用特殊文件夹重定向。

将此设置添加到策略中时，请确保客户端固定驱动器设置存在，且设置为允许。

使用异步写操作

此设置启用或禁用异步磁盘写入。

默认情况下，禁用异步写入。

异步磁盘写入可改善通过广域网执行文件传输和向客户端磁盘写入的速度，此类传输和写入的典型特征是相对较高的带宽以及高延迟。但是，如果发生连接或磁盘故障，正在写入的客户端文件将被置于一种未定义状态。如果发生这种情况，系统将显示一个弹出窗口，告知用户受影响的文件。用户然后可以采取补救措施，例如在重新建立连接时或修复磁盘故障后重新启动中断的文件传输。

Citrix 建议仅为这样的用户启用异步磁盘写入：需要具有良好文件访问速度的远程连接，可以方便地恢复发生连接或磁盘故障时丢失的文件或数据。

将此设置添加到策略中时，请确保客户端驱动器重定向设置存在，且设置为允许。如果禁用此设置，将不执行异步写入。

Flash 重定向策略设置

May 04, 2016

“Flash 重定向”部分包含用于处理用户会话中的 Flash 内容的策略设置。

Flash 加速

此设置允许或禁止 Flash 内容在用户设备上而非在服务器端呈现。默认情况下，启用客户端 Flash 内容呈现。

注意：此设置用于 Citrix Online Plug-in 12.1 的旧版 Flash 重定向功能。

处于启用状态时，此设置会在用户设备上呈现 Flash 内容，从而降低了网络和服务器负载。此外，Flash URL 兼容性列表设置将强制特定 Web 站点上的 Flash 内容在服务器上呈现。

在用户设备上，必须同时启用在用户设备上启用 HDX MediaStream for Flash 设置。

此设置处于禁用状态时，所有 Web 站点（无论 URL 为何）上的 Flash 内容均在服务器上呈现。要仅允许特定 Web 站点上的 Flash 内容在用户设备上呈现，请配置 Flash URL 兼容性列表设置。

Flash 背景色列表

该设置允许您设置指定 URL 的主要颜色。

默认情况下不指定主要颜色。

主要颜色显示在客户端呈现的 Flash 的后面，帮助提供可视区域检测。指定的主要颜色应非常特殊；否则可视区域检测功能可能不会正确执行。

有效条目包含一个 URL（在开头或结尾带有可选的通配符），后接 24 位 RGB 颜色的十六进制代码。例如：<http://citrix.com> 000003。

确保指定的 URL 是 Flash 内容的 URL，可能不同于 Web 站点的 URL。

警告

“注册表编辑器”使用不当会导致严重问题，可能需要重新安装操作系统。Citrix 无法保证因“注册表编辑器”使用不当导致出现的问题能够得以解决。使用“注册表编辑器”需自担风险。请确保在编辑注册表之前进行备份。

在运行 Windows 8 或 Windows 2012 的 VDA 计算机上，此设置可能无法设置 URL 的注册表项颜色。如果出现此情况，请在 VDA 计算机上编辑注册表。

对于 32 位计算机，请使用此注册表设置：

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\HdxMediaStreamForFlash\Server\PseudoServer]
"ForceHDXFlashEnabled"=dword:00000001
```

对于 64 位计算机，请使用此注册表设置：

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer]
"ForceHDXFlashEnabled"=dword:00000001
```

Flash 向后兼容性

此设置将允许或禁止使用早期版本的 Citrix Receiver（以前称为 Citrix 联机插件）具有的原始旧模式 Flash 重定向功能。

默认情况下，此设置处于启用状态。

在用户设备上，必须同时启用在用户设备上启用 HDX MediaStream for Flash 设置。

第二代 Flash 重定向功能处于启用状态，以便与 Citrix Receiver 3.0 结合使用。支持将旧模式重定向功能与 Citrix Online Plug-in 12.1 结合使用。要确保使用第二代 Flash 重定向功能，服务器和用户设备都必须启用第二代 Flash 重定向功能。如果在服务器或用户设备上启用了旧模式重定向功能，则将使用旧模式重定向功能。

Flash 默认行为

此设置将确立第二代 Flash 加速功能的默认行为。

默认情况下启用 Flash 加速。

要配置此设置，请选择以下选项之一：

- 启用 Flash 加速。使用 Flash 重定向。
- 阻止 Flash 播放器。不使用 Flash 重定向和服务器端呈现功能。用户无法查看任何 Flash 内容。
- 禁用 Flash 加速。不使用 Flash 重定向功能。如果服务器上已安装与内容兼容的 Adobe Flash Player for Windows Internet Explorer 版本，则用户可以查看服务器端呈现的 Flash 内容。

对于单个 Web 页面和 Flash 实例，可以覆盖此设置，具体取决于 Flash URL 兼容性列表设置的配置。此外，用户设备必须启用在用户设备上启用 HDX MediaStream for Flash 设置。

Flash 事件日志记录

此设置允许将 Flash 事件记录到 Windows 应用程序事件日志中。

默认情况下，允许记录。

在运行 Windows 7 或 Windows Vista 的计算机上，特定于 Flash 重定向的日志将显示在“应用程序和服务日志”节点下。

Flash 智能回退

如果无需在客户端呈现，或者客户端呈现提供的用户体验较差，则此设置将允许或禁止自动尝试部署服务器端 Flash Player 实例呈现。

默认情况下，此设置处于启用状态。

Flash 延迟阈值

此设置可指定介于 0 到 30 毫秒之间的阈值，以确定 Adobe Flash 内容的呈现位置。

默认情况下，阈值为 30 毫秒。

启动期间，HDX MediaStream for Flash 会测量服务器与用户设备之间的当前延迟。如果延迟低于阈值，则将使用 HDX MediaStream for Flash 将 Flash 内容在用户设备上呈现。如果延迟高于阈值，则在网络服务器上安装了 Adobe Flash 播放器的情况下，Flash 内容将在网络服务器端呈现。

启用此设置时，请确保 Flash 向后兼容设置同样存在，且设置为启用。

注意：仅在旧模式中使用 HDX MediaStream Flash 重定向时适用。

Flash 视频回退预防

此设置指定“小”Flash 内容是否向用户呈现和显示及对应的方式。

默认情况下未配置此设置。

要配置此设置，请选择以下选项之一：

- **仅限小型内容。** 仅在服务器上呈现智能回退内容；其他 Flash 内容将替换为错误 *.swf。
- **仅限使用受支持客户端的小型内容。** 如果客户端当前使用 Flash 重定向，仅在服务器上呈现智能回退内容；其他内容将替换为错误 *.swf。
- **无服务器端内容。** 在服务器上，所有内容均替换为错误 *.swf。

要使用此策略设置，需要指定错误 *.swf 文件。此错误 *.swf 将取代任何您不希望在 VDA 上呈现的内容。

Flash 视频回退预防错误 *.swf

此设置指定错误消息的 URL，使用服务器负载管理时，此错误消息将取代 Flash 实例显示给用户。例如：

`http://domainName.tld/sample/path/error.swf`

Flash 服务器端内容提取 URL 列表

此设置将指定一些 Web 站点，可以将这些站点的 Flash 内容下载到服务器，然后传输到用户设备以进行呈现。

默认情况下，不指定任何站点。

如果用户设备对 Internet 没有直接访问权限，则将使用此设置；服务器将提供 Internet 连接。此外，用户设备必须启用启用服务器端内容提取设置。

第二代 Flash 重定向功能包括回退到在服务器端内容提取 Flash .swf 文件内容。如果用户设备无法从某个 Web 站点提取 Flash 内容，但该 Web 站点在“Flash 服务器端内容提取 URL 列表”中进行了指定，则将自动执行服务器端内容提取。

向该列表中添加 URL 时，请：

- 添加 Flash 应用程序的 URL，而非可启动 Flash Player 的顶层 HTML 页面。
- 在 URL 的开头或结尾处使用星号 (*) 作为通配符。
- 使用后导通配符以包含所有子 URL（例如 `http://www.citrix.com/*`）。
- 显示时使用前缀 `http://` 和 `https://`，但有效列表条目并非必须使用这些前缀。

Flash URL 兼容性列表

此设置将指定一些规则，这些规则决定特定 Web 站点上的 Flash 内容是在用户设备上呈现，在服务器上呈现，还是受到阻止，无法呈现。

默认情况下，不指定任何规则。

向该列表中添加 URL 时，请：

- 排定列表的优先顺序，最重要的 URL、操作和呈现位置在前。
- 在 URL 的开头或结尾处使用星号 (*) 作为通配符。
- 使用后导通配符以包含所有子 URL（例如 `http://www.citrix.com/*`）。
- 显示时使用前缀 `http://` 和 `https://`，但有效列表条目并非必须使用这些前缀。
- 将 Flash 内容无法在用户设备上正确呈现的 Web 站点添加到该列表中，然后选择在服务器上呈现或阻止选项。

图形策略设置

May 04, 2016

“图形”部分包含用于控制如何在用户会话中处理图像的策略设置。

显示内存限制

此设置指定会话的最大视频缓冲区大小 (KB)。

默认情况下，显示内存限制为 65536 KB。

对于需要更高颜色深度和分辨率的连接，可增大该限值。使用如下公式计算所需的最大内存：

内存深度 (字节) = (颜色深度 (bpp) / 8) * (垂直分辨率 [像素]) * (水平分辨率 [像素])。

例如，当颜色深度为 32，垂直分辨率为 600，水平分辨率为 800 时，所需的最大内存为 (32 / 8) * (600) * (800) = 1920000 字节，从而可得显示内存限制为 1920 KB。

只有在已启用旧图形模式策略设置时，才能使用 32 位以外的其他颜色深度。

HDX 仅向每个会话分配所需的显示内存量。因此，如果只有一部分用户所需的内存量高于默认值，通过增加显示内存限制不会对可扩展性产生负面影响。

显示模式降级首选项

注意：对于 Virtual Delivery Agent 7.x，仅当启用旧图形模式策略设置时，才会应用此策略设置。
此设置指定达到会话显示内存限制时的处理方式：首先降低颜色深度，或者首先降低分辨率的级别。

默认情况下，首先降低颜色深度的级别。

达到会话内存限制时，您可以选择首先降低颜色深度还是分辨率的级别，来降低显示图像的质量。如果首先降低颜色深度的级别，显示图像将使用较少的颜色。如果首先降低分辨率的级别，显示图像每英寸将使用较少的像素。

要在颜色深度或分辨率降级时通知用户，请配置在显示模式降级时通知用户设置。

动态窗口预览

该设置启用或禁用以窗口切换、三维窗口切换、任务栏预览和透视窗口预览模式显示无缝窗口。

Windows Aero 预览选项	说明
任务栏预览	用户将鼠标悬停在某个窗口的任务栏图标上时，任务栏上方将显示该窗口的图像。
窗口透视	用户将鼠标悬停在某个任务栏预览图像上时，屏幕上将显示完整大小的窗口图像。
窗口切换	用户按 Alt+Tab 时，系统将为每个打开的窗口显示一个小型预览图标。
三维窗口切换	用户按 Tab+Windows 徽标键时，屏幕上将层叠显示已打开窗口的大图像。

默认情况下，此设置处于启用状态。

图像缓存

注意：对于 Virtual Delivery Agent 7.x，仅当启用旧图形模式策略设置时，才会应用此策略设置。

此设置将在会话中启用或禁用图像分区的缓存和取回。通过在需要时缓存分区中的图像和取回这些图像，用户可以更加顺畅地进行滚动，降低了通过网络传输的数据量，同时降低了需要在用户设备上处理的数据量。

默认启用图像缓存设置。

注意：图像缓存设置控制缓存和取回图像的方式，不控制是否缓存图像。如果启用了“旧图形模式”设置，则将缓存图像。

旧图形模式

此设置将禁用丰富的图形体验并可回退到旧图形体验功能，以改善通过 WAN 或移动连接时的可扩展性。

默认情况下，此设置处于禁用状态，并向用户提供丰富的图形体验。

允许的最大颜色深度

注意：对于 Virtual Delivery Agent 7.x，仅当启用旧图形模式策略设置时，才会应用此策略设置。

此设置指定会话允许的最大颜色深度。

默认情况下，允许的最大颜色深度是每像素 32 位。

此设置仅适用于 ThinWire 驱动程序和连接。不适用于将非 ThinWire 驱动程序用作主显示驱动程序的 VDA，如将 Windows 显示驱动程序模型 (WDDM) 驱动程序用作主显示驱动程序的 VDA。对于将 WDDM 驱动程序用作主显示驱动程序的桌面操作系统 VDA（如 Windows 8），此设置无影响。对于使用 WDDM 驱动程序的 Windows 服务器操作系统 VDA（如 Windows Server 2012 R2），此设置可能会阻止用户连接到 VDA。

设置较高的颜色深度需要更多内存。要在达到内存限制时降低颜色深度的级别，请配置显示模式降级首选项设置。颜色深度降级后，显示图像将使用较少的颜色。

在显示模式降级时通知用户

注意：对于 Virtual Delivery Agent 7.x，仅当启用旧图形模式策略设置时，才会应用此策略设置。

此设置在颜色深度或分辨率降级时，向用户显示简要说明。

默认情况下，禁用用户通知。

排队与丢弃

注意：对于 Virtual Delivery Agent 7.x，仅当启用旧图形模式策略设置时，才会应用此策略设置。

此设置放弃由其他图像替代的排队图像。

默认情况下，启用排队与丢弃。

这将改进向用户设备发送图片时的响应速度。配置此设置会导致由于丢弃帧而使动画断断续续。

缓存策略设置

May 04, 2016

“缓存”部分包含能够在客户端连接的带宽受限时，在用户设备上缓存图像数据的策略设置。

永久缓存阈值

注意：对于 Virtual Delivery Agent 7.x，仅当启用旧图形模式策略设置时，才会应用此策略设置。

此设置用于在用户设备的硬盘驱动器上缓存位图。通过这种方式，可以重复使用之前会话中频繁使用的大型图像。

默认情况下，该阈值为 3000000 bps。

该阈值表示永久性缓存功能生效的上限点。例如，使用默认值时，当带宽低于 3000000 bps 时，将在用户设备的硬盘驱动器上缓存位图。

保持活动状态策略设置

May 04, 2016

“保持活动状态”部分包含用于管理 ICA 保持活动状态消息的策略设置。

ICA 保持活动状态超时

此设置指定相邻 ICA 保持活动状态消息之间间隔的秒数。

默认情况下，保持活动状态消息之间的间隔是 60 秒。

可为 ICA 保持活动状态消息的发送间隔指定 1 到 3600 秒之间的一个值。如果您的网络监视软件负责关闭非活动连接，则不要配置此设置。

ICA 保持活动状态

此设置允许或禁止定期发送 ICA 保持活动状态消息。

默认情况下，不发送 ICA 保持活动状态消息。

启用此设置可防止中断的连接被断开连接。如果服务器检测不到活动，此设置可防止远程桌面服务 (RDS) 断开会话连接。服务器将每隔几秒钟发送一次保持活动状态消息，以检测会话是否处于活动状态。如果会话不再处于活动状态，服务器会将该会话标记为已断开连接。

ICA 保持活动状态在使用会话可靠性时将无效。请仅为不使用会话可靠性的连接配置 ICA 保持活动状态。

相关策略设置：会话可靠性连接。

本地应用程序访问策略设置

May 04, 2016

“本地应用程序访问”部分包含的策略设置可管理在托管桌面环境中用户本地安装的应用程序与托管应用程序的集成。

允许本地应用程序访问

此设置允许或阻止托管桌面环境中用户本地安装的应用程序与托管应用程序的集成。

当用户启动本地安装的应用程序时，即使其实际上是在本地运行，也会看起来是在其虚拟桌面上运行。

默认情况下，禁止本地应用程序访问。

URL 重定向黑名单

此设置指定被重定向到本地 Web 浏览器并在其中启动的 Web 站点。这可能包括需要区域设置信息的 Web 站点（如 msn.com 或 newsgoogle.com），或者包含可更好地呈现在用户设备上的富媒体内容的 Web 站点。

默认情况下，不指定任何站点。

URL 重定向白名单

此设置指定在其启动环境中呈现的 Web 站点。

默认情况下，不指定任何站点。

移动体验策略设置

May 04, 2016

“移动体验”部分包含用于处理 Citrix Mobility Pack 的策略设置。

自动显示键盘

此设置用于启用或禁用移动设备屏幕上的键盘自动显示功能。

默认情况下，键盘的自动显示功能处于禁用状态。

启动经过触控优化的桌面

此设置已禁用，不适用于 Windows 10 计算机。

此设置通过允许或禁止使用为平板电脑设备优化的触控友好界面，控制 Receiver 的整体界面行为。

默认情况下，将使用触控友好界面。

如果仅使用 Windows 界面，请将此策略设置为“禁止”。

远程控制组合框

此设置确定移动设备上的会话中可显示的组合框类型。要显示设备本机组合框控件，请将此策略设置为允许。当此设置为允许时，用户可以将 Receiver for iOS 会话的设置更改为使用 Windows 组合框。

默认情况下，禁止使用远程控制组合框的功能。

多媒体策略设置

May 04, 2016

“多媒体”部分包含用于管理用户会话中的流音频和视频的策略设置。

限制视频质量

此设置指定 HDX 连接允许使用的最大视频质量级别。配置后，最大视频质量将限制为指定值，确保在环境中保持多媒体服务质量 (QoS)。

默认情况下未配置此设置。

要限制允许使用的最大视频质量级别，请选择以下任一选项：

- 1080 p/8.5 mbps
- 720 p/4.0 mbps
- 480 p/720 kbps
- 380 p/400 kbps
- 240 p/200 kbps

注意：在同一台服务器上同时播放多个视频会消耗大量资源，并且可能影响服务器的可扩展性。

多媒体会议

此设置启用或禁用对视频会议应用程序的支持。

默认情况下，允许视频会议支持。

将此设置添加到策略时，请确保 Windows Media 重定向设置存在，且设置为允许。

使用多媒体会议时，请确保满足以下条件：

- 必须安装制造商为用于多媒体会议的网络摄像机提供的驱动程序。
- 必须先将网络摄像机连接到用户设备，然后再启动视频会议会话。服务器在任何给定的时间只使用一个已安装的网络摄像机。如果用户设备上安装了多个网络摄像机，服务器会依次尝试使用每个网络摄像机，直至成功创建视频会议会话。

通过 WAN 优化 Windows Media 多媒体重定向

此设置启用了实时多媒体代码转换，允许通过流技术将音频和视频媒体推送到移动设备，并利用改善通过 WAN 交付 Windows Media 内容的方式增强了用户体验。

默认情况下，已优化通过 WAN 的 Windows Media 内容交付。

将此设置添加到策略时，请确保 Windows Media 重定向设置存在，且设置为允许。

启用此设置后，将根据需要自动部署实时多媒体代码转换以启用媒体流，因此即使在恶劣的网络条件下也可以提供无缝用户体验。

使用 GPU 以通过 WAN 优化 Windows Media 多媒体重定向

此设置支持在 Virtual Delivery Agent (VDA) 上的图形处理器 (GPU) 中执行实时多媒体代码转换，以改善服务器可扩展性。仅在 VDA 具有支持硬件加速的 GPU 时，GPU 代码转换才可用。否则，代码转换将回退到 CPU。

注意：只有 NVIDIA GPU 才支持 GPU 代码转换。

默认情况下，禁止使用 VDA 上的 GPU 通过 WAN 优化 Windows Media 内容交付。

将此设置添加到策略后，请确保 Windows Media 重定向和在 WAN 中对 Windows Media 多媒体重定向进行优化设置存在，且设置为允许。

视频回退预防

管理员可以使用视频回退预防策略设置指定向用户交付流内容时尝试使用的方法。

默认情况下未配置此设置。这允许客户端提取到 RAVE 再到服务器端回退。

要配置此设置，请选择以下选项之一：

- **服务器提取 - 服务器端呈现。** 允许客户端提取到 RAVE 再到服务器端回退。
- **服务器提取 - 客户端呈现。** 允许客户端提取到 RAVE 回退，但阻止 RAVE 到服务器端呈现回退。
- **客户端提取 - 客户端呈现。** 阻止客户端提取到 RAVE 再到服务器端呈现回退。

内容不播放时，播放器窗口会显示错误消息“由于缺少资源，因此公司阻止了该视频”。

Windows Media 客户端内容提取

此设置使用户设备能够通过流技术直接从 Internet 或 Intranet 上的源提供程序推送多媒体文件，而非通过主机服务器推送。

默认情况下，允许通过流技术将多媒体文件直接从源提供程序推送到用户设备。

启用此设置可将对媒体的任何处理过程从主机服务器移动到用户设备，提高了网络利用率和服务器可扩展性。此外，启用此设置后，将不需要在用户设备上安装 Microsoft DirectShow 或媒体基础等高级多媒体框架；用户设备只需要能够播放 URL 中的文件。

将此设置添加到策略时，请确保 Windows Media 重定向设置存在，且设置为允许。如果禁用此设置，将禁用通过流技术将多媒体文件直接从源提供程序推送到用户设备。

Windows Media 重定向

此设置可控制和优化服务器向用户交付流音频和视频的方式。

默认情况下，允许向用户交付流音频和视频。

启用此设置可将服务器上呈现的音频和视频质量提高到一个级别，可与用户设备上本地播放的音频和视频质量相媲美。服务器会将多媒体以原始压缩格式通过流技术推送到客户端，并允许用户设备解压缩和呈现该媒体。

Windows Media 重定向可优化使用编解码器编码的多媒体文件，这些编解码器遵循 Microsoft DirectShow、DirectX 媒体对象 (DMO) 和媒体基础标准。要播放给定的多媒体文件，用户设备上必须存在与多媒体文件的编码格式兼容的编解码器。

默认情况下，在 Citrix Receiver 上禁用音频。要允许用户在 ICA 会话中运行多媒体应用程序，请打开音频或授予用户在其 Receiver 接口上自行打开音频的权限。

仅当使用 Windows Media 重定向播放媒体的效果比使用基本 ICA 压缩和常规音频所呈现的效果差时，才选择已禁止。这种情况很少见，但在带宽较低的情况下可能会发生。例如，当播放关键帧的频率非常低的媒体时。

Windows Media 重定向缓冲区大小

此设置为多媒体加速指定 1 到 10 秒的缓冲区大小。

默认情况下，缓冲区大小为 5 秒。

Windows Media 重定向缓冲区大小的使用

该设置启用或禁用使用 Windows Media 重定向缓冲区大小设置中所指定的缓冲区大小。

默认情况下，不使用指定的缓冲区大小。

如果禁用此设置，或如果未配置“Windows Media 重定向缓冲区大小”设置，服务器将使用默认缓冲区大小值（5 秒）。

多流连接策略设置

May 04, 2016

“多流连接”部分中包含的策略设置可用于在一个会话中管理多个 ICA 连接的服务质量 (QoS) 优先级顺序。

通过 UDP 传输音频

此设置允许或阻止通过 UDP 在服务器上传输音频。

默认情况下，允许通过 UDP 在服务器上传输音频。

启用后，此设置在服务器上打开一个 UDP 端口以支持配置为使用“通过 UDP 实时传输音频”的所有连接。

音频 UDP 端口范围

此设置指定 Virtual Delivery Agent (VDA) 用于与用户设备交换音频数据包数据的端口号范围（采用最小端口号.最大端口号格式）。VDA 尝试使用每个 UDP 端口对与用户设备交换数据，从最小端口号开始尝试，之后每尝试一次，端口号增加 2。每个端口可同时处理入站和出站通信。

默认情况下，此范围设置为 16500,16509。

多端口策略

此设置指定用于 ICA 通信的 TCP 端口并为每个端口建立网络优先级。

默认情况下，主端口 (2598) 拥有“高”优先级。

配置端口时，可以分配以下优先级：

- 很高 - 适用于实时活动，例如视频会议
- 高 - 适用于交互元素，例如屏幕、键盘和鼠标
- 中 - 适用于批量进程，例如客户端驱动器映射
- 低 - 适用于后台活动，例如打印

每个端口必须有唯一的优先级。例如，不能同时为 CGP 端口 1 和 CGP 端口 3 分配“很高”优先级。

要从优先级顺序中删除某个端口，请将其端口号设置为 0。您无法删除主端口，也无法修改其优先级。

配置此设置时，请重新启动服务器。只有启用了多流计算机设置策略设置时，此设置才会生效。

多流计算机设置

此设置可在服务器上启用或禁用“多流”功能。

默认情况下，“多流”功能处于禁用状态。

如果要在环境中使用具有“多流”支持功能的 Citrix CloudBridge，则无需配置此设置。如果要使用第三方路由器或旧版 Branch Repeater 实现所需的服务质量 (QoS)，应配置此策略。

配置此策略时，应重新启动服务器以确保所做的更改生效。

重要：将此策略设置与带宽限制策略设置（例如总会话带宽限制）结合使用可能会产生意外结果。如果要在策略中包含此设置，请确保将带宽限制设置排除在外。

多流用户设置

此设置可在用户设备上启用或禁用“多流”功能。

默认情况下，对所有用户禁用“多流”功能。

只有在启用了多流计算机设置策略设置的主机上，此设置才会生效。

重要：将此策略设置与带宽限制策略设置（例如总会话带宽限制）结合使用可能会产生意外结果。如果要在策略中包含此设置，请确保将带宽限制设置排除在外。

端口重定向策略设置

May 04, 2016

端口重定向部分包含用于客户端 LPT 和 COM 端口映射的策略设置。

注意：对于 Virtual Delivery Agent 7.x，请使用注册表配置这些设置。请参阅[使用注册表配置 COM 端口和 LPT 端口重定向设置](#)。

自动连接客户端 COM 端口

此设置启用或禁用用户登录到站点时用户设备上 COM 端口的自动连接。

默认情况下，不自动连接客户端 COM 端口。

自动连接客户端 LPT 端口

此设置启用或禁用用户登录到站点时用户设备上 LPT 端口的自动连接。

默认情况下，不自动连接客户端 LPT 端口。

客户端 COM 端口重定向

此设置允许或阻止访问用户设备上的 COM 端口。

默认情况下，禁止 COM 端口重定向。

下列策略设置为相关设置：

- COM 端口重定向带宽限制
- COM 端口重定向带宽限制百分比

客户端 LPT 端口重定向

此设置允许或阻止访问用户设备上的 LPT 端口。

默认情况下，禁止 LPT 端口重定向。

只有将打印作业发送至 LPT 端口（而非发送至用户设备上的打印对象）的旧版应用程序才可使用 LPT 端口。目前大多数应用程序都可将打印作业发送至打印机对象。只有用于托管打印至 LPT 端口的旧版应用程序的服务器，才有必要使用此策略设置。

下列策略设置为相关设置：

- LPT 端口重定向带宽限制
- LPT 端口重定向带宽限制百分比

打印策略设置

May 04, 2016

“打印”部分包含用于管理客户端打印的策略设置。

客户端打印机重定向

此设置控制在用户登录到会话时客户端打印机是否映射到服务器。

默认情况下，允许客户端打印机映射。如果禁用此设置，则不会自动创建会话的 PDF 打印机。

相关策略设置：自动创建客户端打印机

默认打印机

此设置指定在会话中如何在用户设备上建立默认打印机。

默认情况下，用户的当前打印机用作会话的默认打印机。

要为默认打印机使用当前的远程桌面服务或 Windows 用户配置文件设置，请选择不调整用户的默认打印机。如果选择此选项，默认打印机将不会保存在用户配置文件中，并且不会随其他会话或客户端属性而改变。会话中的默认打印机将是该会话中自动创建的第一个打印机，可以是：

- 第一台通过控制面板 > 设备和打印机添加到 Windows 服务器的本地打印机。
- 第一台自动创建的打印机（如果没有向服务器添加任何本地打印机）。

可以使用此选项通过配置文件设置向用户呈现最近的打印机（即邻近打印）。

打印机分配

此设置是默认打印机和会话打印机设置的一个替代方案。使用单独的默认打印机和会话打印机设置对站点、大型组或组织单位的行为进行配置。使用打印机分配设置，可以将大型的打印机组分配给多个用户。

此设置指定在会话中如何在所列的用户设备上建立默认打印机。

默认情况下，用户的当前打印机用作会话的默认打印机。

该设置还指定了在会话中将为每个用户设备自动创建的网络打印机。默认情况下，不指定任何打印机。

- 在设置默认打印机的值时：
要使用用户设备当前默认的打印机，请选择不调整。

要使用默认打印机当前的远程桌面服务或 Windows 用户配置文件设置，请选择不调整。如果选择此选项，默认打印机将不会保存在用户配置文件中，并且不会随其他会话或客户端属性而改变。会话中的默认打印机将是该会话中自动创建的第一个打印机，可以是：

- 第一台通过控制面板 > 设备和打印机添加到 Windows 服务器的本地打印机。
- 第一台自动创建的打印机（如果没有向服务器添加任何本地打印机）。
- 设置会话打印机值时：添加打印机，键入要自动创建的打印机的 UNC 路径。添加打印机后，可以在每次登录时为当前会话应用自定义设置。

打印机自动创建事件日志首选项

此设置指定在打印机自动创建过程中记录哪些事件。您可以选择不记录错误或警告，只记录错误，或同时记录错误和警告。

默认情况下，将记录错误和警告。

以下事件就是警告的一个例子：未能安装打印机的本机驱动程序，而是安装了通用打印驱动程序。要在此案例中使用通用打印驱动程序，可将通用打印驱动程序用法设置配置为仅使用通用打印或仅当请求的驱动程序不可用时才使用通用打印。

会话打印机

此设置指定在会话中将自动创建的网络打印机。

默认情况下，不指定任何打印机。

要添加打印机，请键入要自动创建的打印机的 UNC 路径。添加打印机后，可以在每次登录时为当前会话应用自定义设置。

等待创建打印机（服务器桌面）

此设置允许或阻止连接到会话时发生延迟，此延迟便于自动创建服务器桌面打印机。

默认情况下不发生连接延迟。

客户端打印机策略设置

May 04, 2016

“客户端打印机”部分包含用于客户端打印机的策略设置（包括自动创建客户端打印机、保留打印机属性以及连接到打印服务器的设置）。

自动创建客户端打印机

此设置指定自动创建的客户端打印机。此设置可覆盖默认的客户端打印机自动创建设置。

默认情况下，所有客户端打印机都是自动创建的。

仅当客户端打印机重定向设置存在，且设置为允许时，此设置才能生效。

将此设置添加到策略时，请选择以下选项之一：

- 自动创建所有客户端打印机可在用户设备上自动创建所有打印机。
- 仅自动创建客户端的默认打印机仅自动创建选择作为用户设备上的默认打印机的打印机。
- 仅自动创建本地(非网络)客户端打印机将仅自动创建通过 LPT、COM、USB、TCP/IP 或其他本地端口直接连接到用户设备的打印机。
- 不自动创建客户端打印机在用户登录时关闭所有客户端打印机的自动创建功能。这会导致在优先级较低的策略中，以自动创建客户端打印机的远程桌面服务 (RDS) 设置覆盖此设置。

自动创建一般通用打印机

注意：用于解决此策略设置问题的修补程序以知识库文章 CTX141565 和 CTX141566 的方式提供。
该设置为所使用的用户设备与通用打印兼容的会话启用或禁用一般 Citrix 通用打印机对象的自动创建。

默认情况下不自动创建一般通用打印机对象。

下列策略设置为相关设置：

- 通用打印驱动程序使用
- 通用驱动程序首选项

客户端打印机名称

此设置为自动创建的客户端打印机选择命名约定。

默认情况下，使用标准打印机名称。

选择标准打印机名称以使用类似“HPLaserJet 4 from clientname in session 3”的打印机名称。

选择旧版打印机名称可让使用 MetaFrame Presentation Server 3.0 或较旧版本的用户或组使用旧版客户端打印机名称并保留向后兼容性。旧版打印机名称示例：“Client/clientname#/HPLaserJet 4”。此选项不够安全。

注意：仅提供此选项以用于向后兼容旧版 XenApp 和 XenDesktop。

直接连接到打印服务器

此设置为可访问的网络共享上托管的客户端打印机启用或禁用从托管应用程序的虚拟桌面或服务器到打印服务器的直接连接。

默认情况下，启用直接连接。

如果托管应用程序的虚拟桌面或服务器未通过 WAN 与网络打印服务器连接，请启用直接连接。如果网络打印服务器和托管应用程序的虚拟桌面或服务器位于相同的 LAN，那么直接通信可加快打印速度。

如果网络通过 WAN、延迟较大或者带宽有限，请禁用直接连接。打印作业通过将其重定向到网络打印服务器的用户设备进行路由。发送到用户设备的数据会经过压缩，因此通过 WAN 传输数据会占用较少的带宽。

如果存在两台同名的网络打印机，则会使用与用户设备位于相同网络的打印机。

打印机驱动程序映射和兼容性

此设置为自动创建的客户端打印机指定驱动程序替换规则。

默认情况下，不指定任何规则。

定义驱动程序替代规则时，可以允许或禁止使用指定的驱动程序创建打印机。此外，可以允许创建的打印机仅使用通用打印驱动程序。驱动程序替换将覆盖或映射用户设备提供的打印机驱动程序名称，从而替换服务器上的等效驱动程序。这样可使服务器应用程序有权访问与服务器具有相同驱动程序，但驱动程序名称不同的客户端打印机。

可以添加驱动程序映射，编辑现有映射，覆盖映射的自定义设置，删除映射或更改驱动程序条目在列表中的顺序。添加映射时，请输入客户端打印机驱动程序名称，然后选择要替换的服务器驱动程序。

打印机属性保留

此设置指定是否存储打印机属性以及打印机属性的存储位置。

默认情况下，系统会决定是将打印机属性存储在用户设备上（如果有），还是存储在用户配置文件中。

将此设置添加到策略时，请选择以下选项之一：

- 仅保存在客户端设备上适用于拥有不保存的强制配置文件或漫游配置文件的用户设备。仅当场中的所有服务器都运行 XenApp 5 及更高版本，并且用户使用 Citrix Online Plug-in 9 至 12.x 版或使用 Citrix Receiver 3.x 时，才选择该选项。
- 仅保留在用户配置文件中适用于受带宽（此选项会减少网络流量）和登录速度限制的用户设备，或适用于使用旧插件的用户。此选项将打印机属性存储在服务器上的用户配置文件中，并阻止与用户设备交换任何属性。如果使用 MetaFrame Presentation Server 3.0 或较旧版本和 MetaFrame Presentation Server Client 8.x 或较旧版本，请使用此选项。请注意，此选项仅在使用远程桌面服务 (RDS) 漫游配置文件时适用。
- 仅当未保存在客户端时才保留在配置文件中允许系统决定打印机属性的存储位置。打印机属性会存储在用户设备上（如果有）或用户配置文件中。虽然此选项最为灵活，但也会延长登录时间，且需要使用额外的带宽执行系统检查。
- 不保留打印机属性将阻止存储打印机属性。

保留和恢复的客户端打印机

此设置启用或禁用用户设备上的打印机的保留和重新创建。默认情况下，客户端打印机将自动保留和自动恢复。

保留的打印机属于用户创建的打印机，在下一个会话启动时会再次创建（或被记住）。XenApp 重新创建保留的打印机时，它会考虑使用除自动创建客户端打印机设置以外的所有策略设置。

恢复的打印机属于管理员完全自定义的打印机，其保存状态为永久连接到客户端端口。

驱动程序策略设置

May 04, 2016

“驱动程序”部分包含与打印机驱动程序有关的策略设置。

自动安装现成的打印机驱动程序

该设置启用或禁用从 Windows 现成驱动程序集或从在主机上分段的驱动程序包（使用 pnputil.exe /a）来自动安装打印机驱动程序。

默认情况下，会根据需要安装这些驱动程序。

通用驱动程序首选项

此设置指定使用通用打印机驱动程序的顺序，从列表中的第一个条目开始。

默认情况下，首选顺序为：

- EMF
- XPS
- PCL5c
- PCL4
- PS

您可以在列表中添加、编辑或删除驱动程序，以及更改驱动程序的顺序。

通用打印驱动程序使用

此设置指定何时使用通用打印。

默认情况下，仅当请求的驱动程序不可用时才使用通用打印。

通用打印使用一般打印机驱动程序取代标准的特定于打印机型号的驱动程序，从而潜在地减轻了在主计算机上管理驱动程序的负担。通用打印驱动程序的可用性取决于用户设备、主机和打印服务器软件的功能。在某些配置中，通用打印可能不可用。

将此设置添加到策略时，请选择以下选项之一：

- 仅使用特定于打印机型号的驱动程序指定客户端打印机仅使用在登录时自动创建的特定于打印机型号的标准驱动程序。如果请求的驱动程序不可用，将无法自动创建客户端打印机。
- 仅使用通用打印指定不使用特定于打印机型号的标准驱动程序。仅使用通用打印驱动程序创建打印机。
- 仅当请求的驱动程序不可用时才使用通用打印使用标准的特定于打印机型号的驱动程序来创建打印机（如果这些驱动程序可用）。如果该驱动程序在服务器上不可用，则使用合适的通用驱动程序自动创建客户端打印机。
- 仅当通用打印不可用时才使用打印机型号专用的驱动程序在通用打印驱动程序可用时将使用此驱动程序。如果该驱动程序在服务器上不可用，则使用合适的特定于打印机型号的驱动程序来自动创建客户端打印机。

通用打印服务器策略设置

May 04, 2016

“通用打印服务器”部分包含用于处理通用打印服务器的策略设置。

启用通用打印服务器

此设置启用或禁用托管应用程序的虚拟桌面或服务器上的“通用打印服务器”功能。此策略设置适用于包含托管应用程序的虚拟桌面或服务器的组织单位 (OU)。

默认情况下，通用打印服务器处于禁用状态。

将此设置添加到策略时，请选择以下选项之一：

- **启用并允许回退到 Windows 本机远程打印。**通用打印服务器将在可能的情况下提供网络打印机连接服务。如果通用打印服务器不可用，将使用 Windows 打印提供程序。Windows 打印提供程序将继续处理之前使用 Windows 打印提供程序创建的所有打印机。
- **启用但不允许回退到 Windows 本机远程打印。**通用打印服务器将独自提供网络打印机连接服务。如果通用打印服务器不可用，网络打印机连接将失败。此设置可以有效禁用通过 Windows 打印提供程序进行的网络打印。当启用了包含此设置的策略时，将不会创建之前曾使用 Windows 打印提供程序创建的打印机。
- **已禁用。**禁用通用打印服务器功能。在连接到具有 UNC 名称的网络打印机时，不会尝试连接通用打印服务器。与远程打印机的连接将继续使用 Windows 本机远程打印工具。

通用打印服务器打印数据流(CGP)端口

此设置指定通用打印服务器的打印数据流通用网关协议 (CGP) 侦听器使用的 TCP 端口号。此策略设置仅适用于包含打印服务器的 OU。

默认情况下，此端口号设置为 7229。

有效的端口号必须在 1 到 65535 范围内。

通用打印服务器打印流输入带宽限制 (kpbs)

此设置指定每个打印作业使用 CGP 向通用打印服务器传输打印数据的速率上限值 (Kbps)。此策略设置适用于包含托管应用程序的虚拟桌面或服务器的 OU。

默认值为 0，表示不指定上边界。

通用打印服务器 Web 服务(HTTP/SOAP)端口

此设置指定通用打印服务器 Web 服务 (HTTP/SOAP) 侦听器所使用的 TCP 端口。通用打印服务器是一个可选组件，允许将 Citrix 通用打印驱动程序用于网络打印场景。在使用通用打印服务器时，打印命令将从 XenApp 和 XenDesktop 主机通过 SOAP over HTTP 发送到通用打印服务器。此设置将修改通用打印服务器侦听传入的 HTTP/SOAP 请求所使用的默认 TCP 端口。

必须配置相同的主机和打印服务器 HTTP 端口。如果配置的端口不相同，主机软件将连接不到通用打印服务器。此设置更改 XenApp 和 XenDesktop 上的 VDA。此外，还必须更改通用打印服务器上的默认端口。

默认情况下，此端口号设置为 8080。

有效的端口号必须在 0 到 65535 范围内。

通用打印策略设置

May 04, 2016

“通用打印”部分包含用于管理通用打印的策略设置。

通用打印 EMF 处理模式

该设置控制在 Windows 用户设备上处理 EMF 后台打印文件的方法。

默认情况下，系统将 EMF 记录直接后台处理到打印机中。

将此设置添加到策略时，请选择以下选项之一：

- 为打印机重新处理 EMF 强制重新处理 EMF 后台打印文件，并通过用户设备上的 GDI 子系统发送。可以将该设置用于需要重新处理 EMF 但在会话中可能未自动选择这样执行的驱动程序。
- 直接后台打印到打印机，与 Citrix 通用打印驱动程序一起使用时，确保 EMF 记录后台打印并交付到用户设备进行处理。通常，这些 EMF 后台打印文件直接插入到客户端的后台打印队列中。对于与 EMF 格式兼容的打印机和驱动程序，这是速度最快的打印方法。

通用打印图像压缩限制

此设置指定通过 Citrix 通用打印驱动程序打印的图像的最高质量和最低压缩级别。

默认情况下，图像压缩限制设置为最佳质量(无损压缩)。

如果选择无压缩，将仅对 EMF 打印禁用压缩。

将此设置添加到策略时，请选择以下选项之一：

- 不压缩
- 最佳质量(无损压缩)
- 高质量
- 标准质量
- 降低质量(最大压缩)

将该设置添加到包含通用打印优化默认设置的策略中时，应注意以下几项：

- 如果通用打印图像压缩限制设置中的压缩级别低于通用打印优化默认值设置中所定义的级别，将按照通用打印图像压缩限制设置中所定义的级别对图像进行压缩。
- 如果禁用压缩，则通用打印优化默认值设置的所需图像质量和启用超级压缩选项在策略中不起作用。

通用打印优化默认值

该设置指定为会话创建通用打印驱动程序时打印优化的默认值。

- 所需图像质量指定应用到通用打印的默认图像压缩限制。默认情况下，标准质量处于启用状态，这意味着用户只能使用标准或降低质量的压缩级别来打印图像。
- 启用超级压缩用于启用或禁用超出由所需图像质量所设置的压缩级别时减少带宽，而不降低图像质量。默认情况下，超级压缩功能处于禁用状态。
- 图像与字体缓存设置指定是否缓存在打印流中多次出现的图像和字体，以确保每个唯一的图像或字体只发送给打印机一次。默认情况下将缓存嵌入式图像和字体。请注意，只有在用户设备支持此行为时，这些设置才适用。
- 允许非管理员修改这些设置指定用户是否可以更改会话内的默认打印优化设置。默认情况下，不允许用户更改默认打印优化设置。

注意：EMF 打印支持所有这些选项。对于 XPS 打印，则仅支持所需图像质量选项。

将该设置添加到包含通用打印图像压缩限制设置的策略中时，应注意以下几项：

- 如果通用打印图像压缩限制设置中的压缩级别低于通用打印优化默认值设置中所定义的级别，将按照通用打印图像压缩限制设置中所定义的级别对图像进行压缩。
- 如果禁用压缩，则通用打印优化默认值设置的所需图像质量和启用超级压缩选项在策略中不起作用。

通用打印预览首选项

此设置指定是否对自动创建的打印机或一般通用打印机使用打印预览功能。

默认情况下，不对自动创建的打印机或一般通用打印机使用打印预览。

将此设置添加到策略时，请选择以下选项之一：

- 不对自动创建的打印机或一般通用打印机使用打印预览
- 仅对自动创建的打印机使用打印预览
- 仅对一般通用打印机使用打印预览
- 对自动创建的打印机和一般通用打印机都使用打印预览

通用打印打印质量限制

此设置指定在会话中生成打印输出时可用的最高分辨率 (dpi)。

默认情况下，无限制处于启用状态，这意味着用户可以选择其连接的打印机所允许的最高打印质量。

如果配置此设置，它将在输出分辨率方面限制用户可以达到的最高打印质量。打印质量本身和用户所连接的打印机的打印质量能力限制为已配置的设置。例如，如果打印质量设置配置为中分辨率(600 DPI)，则用户打印输出限制为最高质量为 600 DPI，“通用打印机”对话框“高级”选项卡中的“打印质量”设置显示的分辨率设置最高只能达到“中质量(600 DPI)”。

将此设置添加到策略时，请选择以下选项之一：

- 草稿 (150 DPI)
- 低分辨率 (300 DPI)
- 中分辨率 (600 DPI)
- 高分辨率 (1200 DPI)
- 无限制

安全策略设置

May 04, 2016

“安全”部分介绍了配置会话加密和登录数据加密的相关策略设置。

SecureICA 最低加密级别

此设置指定服务器与用户设备之间所传输会话数据的最低加密级别。

重要提示：

对于 Virtual Delivery Agent 7.x，只能使用此策略设置来启用通过“RC5 (128 位)”加密实现的登录数据加密。其他设置仅在用于向后兼容旧版 XenApp 和 XenDesktop 时提供。

对于 VDA 7.x，使用 VDA 交付组的基本设置来设置会话数据的加密。如果为交付组选择了“启用安全 ICA”，会话数据将通过“RC5 (128 位)”加密进行加密。如果没有为交付组选择“启用安全 ICA”，会话数据将通过“基本”加密进行加密。

将此设置添加到策略时，请选择以下选项之一：

- 基本可使用一种非 RC5 算法加密客户端连接。它保护数据流使之不能被直接读取，但可以解密。默认情况下，服务器对客户端-服务器通信流使用基本加密。
- 仅限 RC5 (128 位) 登录使用 RC5 128 位加密来加密登录数据，使用基本加密来加密客户端连接。
- RC5 (40 位) 使用 RC5 40 位加密来加密客户端连接。
- RC5 (56 位) 使用 RC5 56 位加密来加密客户端连接。
- RC5 (128 位) 使用 RC5 128 位加密来加密客户端连接。

为客户端-服务器加密指定的设置可能会与您的环境和 Windows 操作系统中的任何其他加密设置进行交互。如果在服务器或用户设备上设置了优先级更高的加密级别，则您为已发布的资源指定的设置可能会被覆盖。

您可以为特定用户提高加密级别，以进一步加强其通信安全和消息的完整性。如果某项策略需要更高的加密级别，则使用较低加密级别的 Receiver 将被拒绝连接。

SecureICA 不执行身份验证，也不检查数据完整性。要为站点提供端到端加密，请将 SecureICA 与 SSL/TLS 加密一起使用。

SecureICA 不使用符合 FIPS 标准的算法。如果这会带来问题，请将服务器和 Receiver 配置为使用 SecureICA。

服务器限制策略设置

May 04, 2016

“服务器限制”部分包含用于控制空闲连接的策略设置。

服务器空闲计时器间隔

此设置确定如果用户未输入任何内容，用户会话可以保持多久不中断（以毫秒为单位）。

默认情况下，空闲连接不会断开连接（服务器空闲计时器间隔 = 0）。

注意

使用此策略设置时，如果会话空闲超过指定的时间，可能会向用户显示“空闲计时器已过期”对话框。这是 Microsoft 对话框，不受 Citrix 策略设置控制。有关详细信息，请参阅 <http://support.citrix.com/article/CTX118618>。

会话限制策略设置

May 04, 2016

“会话限制”部分包含的策略设置可用于控制会话在强制注销前可保持连接的时间长度。

断开连接的会话计时器

此设置启用或禁用计时器，用于指定断开连接的锁定桌面在注销会话之前可以保持锁定状态的时间长度。

默认情况下，断开连接的会话不注销。

断开连接的会话计时器间隔

此设置用于指定断开连接的锁定桌面在注销会话前可保持锁定状态的时间长度（分钟）。

默认情况下，此时间段为 1440 分钟（24 小时）。

会话连接计时器

此设置启用或禁用计时器，用于指定用户设备与桌面之间实现不间断连接的最长持续时间。

默认情况下，禁用此计时器。

会话连接计时器间隔

此设置用于指定用户设备与桌面之间实现不间断连接的最长持续时间（分钟）。

默认情况下，最长持续时间为 1440 分钟（24 小时）。

会话空闲计时器

此设置启用或禁用计时器，用于指定在没有用户输入的情况下用户设备与桌面之间实现不间断连接的时间长度。

默认情况下，启用此计时器。

会话空闲计时器间隔

此设置用于指定如果用户不输入任何内容，用户设备与桌面的连接可以保持多久不中断。

默认情况下，空闲连接将保持 1440 分钟（24 小时）。

会话可靠性策略设置

May 04, 2016

“会话可靠性”部分包含用于管理会话可靠性连接的策略设置。

会话可靠性连接

此设置允许或阻止会话在失去网络连接期间保持打开状态。

默认情况下，允许启用会话可靠性。

会话可靠性使会话在网络连接中断时保持活动状态并显示在用户的屏幕上。用户仍然可以看到他们正在使用的应用程序，直至网络连接恢复。

会话可靠性可使会话在服务器上保持活动状态。要指示连接已断开，用户的显示内容将冻结，且光标变成一个旋转的沙漏，直至连接恢复。用户在连接中断期间可继续访问显示内容，在网络连接恢复后可继续与应用程序交互。会话可靠性可重新连接用户而不提示进行重新身份验证。如果您不希望用户无需重新进行身份验证即可重新连接到已中断的会话，请将客户端自动重新连接身份验证设置配置为要求进行身份验证。这样，在重新连接到中断的会话时，将提示用户重新进行身份验证。

如果您同时使用了会话可靠性和客户端自动重新连接，这两项功能将按顺序发挥作用。会话可靠性将在会话可靠性超时设置中指定的时间之后关闭（或断开）用户会话。之后，客户端自动重新连接策略设置生效，尝试将用户重新连接到断开连接的会话。

会话可靠性端口号

此设置为传入会话可靠性连接指定 TCP 端口号。

默认情况下，此端口号设置为 2598。

会话可靠性超时

此设置指定会话可靠性代理等待用户重新连接的时间长度（以秒为单位），达到该时间后将允许会话断开连接。

默认情况下，此时间设置为 180 秒或 3 分钟。

尽管您可以延长会话保持打开状态的时间长度，此功能的主要目的是为用户提供方便，它不会提示用户重新进行身份验证。但是，如果您延长会话保持打开状态的时间，则可能导致用户感到不耐烦而离开用户设备，从而使未经授权的用户有机会访问该会话。

时区控制策略设置

May 04, 2016

“时区控制”部分包含与在会话中使用本地时间相关的策略设置。

估算旧版客户端的本地时间

此设置启用或禁用对向服务器发送了不准确时区信息的用户设备进行的本地时区估算。

默认情况下，服务器在必要时将估算本地时区。

此设置旨在用于不向服务器发送时区详细信息的旧版 Receivers 或 ICA 客户端。用于向服务器发送时区详细信息的 Receiver（如支持的 Receiver for Windows 版本）时，此设置不起作用。

使用客户端本地时间

此设置确定用户会话的时区设置。可以是用户会话的时区，也可以是用户设备的时区。

默认情况下，使用用户会话的时区。

为使此设置生效，请在组策略编辑器中启用允许时区重定向设置（“用户配置”>“管理模板”>“Windows 组件”>“远程桌面服务”>“远程桌面会话主机”>“设备和资源重定向”）。

TWAIN 设备策略设置

May 04, 2016

“TWAIN 设备”部分包含的策略设置与以下内容相关：映射客户端 TWAIN 设备（例如数码相机或扫描仪），优化从服务器到客户端的图像传输。

注意

当前不支持 TWAIN 2.0。

客户端 TWAIN 设备重定向

此设置允许或禁止用户从服务器上托管的图像处理应用程序访问用户设备上的 TWAIN 设备。默认情况下，允许 TWAIN 设备重定向。

下列策略设置为相关设置：

- TWAIN 压缩级别
- TWAIN 设备重定向带宽限制
- TWAIN 设备重定向带宽限制百分比

TWAIN 压缩级别

此设置指定从客户端到服务器的图像传输的压缩级别。低可提供最佳图像质量，中可提供良好图像质量，高可提供低图像质量。默认情况下，应用中级压缩。

USB 设备策略设置

May 04, 2016

“USB 设备”部分包含用于管理 USB 设备文件重定向的策略设置。

客户端 USB 设备优化规则

在 XenApp 和 XenDesktop 7.6 FP3 中，可以向设备应用客户端 USB 设备优化规则以禁用优化，或更改优化模式。

用户插入 USB 输入设备时，主机检查 USB 策略设置是否允许此设备。如果不允许此设备，主机则检查此设备的**客户端 USB 设备优化规则**。如果未指定任何规则，将按照交互模式 (02) 处理设备。捕获模式 (04) 是签名设备的建议模式。请参阅下面的说明以了解可用模式。

须知事项

- 对于使用 Wacom 签名板和平板电脑的情况，我们建议禁用屏幕保护程序。本部分结尾介绍执行此操作的步骤。
- 安装 XenApp 和 XenDesktop 7.6 FP3 的 XenApp 和 XenDesktop 策略时已经预配置了对优化 Wacom STU 签名板和平板电脑系列的支持。
- 签名设备在整个 XenApp 和 XenDesktop 中均可以使用，且无需使用驱动程序作为签名设备。Wacom 包含可以安装以进一步自定义设备的其他软件。请参阅 <http://www.wacom.com/>。
- 手写板。某些手写输入设备在 PCI/ACPI 总线上可能显示为 HID 设备，不受支持。这些设备应该连接到客户端上的 USB 主机控制器，以便在 XenDesktop 会话内重定向。

策略规则采用以空格分隔的 tag=value 表达式格式。支持以下标记：

标记名称	说明
模式	为输入设备支持优化模式，类 =03。支持的模式包括： 无优化 - 值 01。 交互模式 - 值 02。手写板和 3D 专业鼠标等设备的建议模式。 捕获模式 - 值 04。签名板等设备的首选模式。
VID	设备描述符中的供应商 ID
PID	设备描述符中的产品 ID
REL	设备描述符中的版本 ID
Class	设备描述符或接口描述符中的类
SubClass	设备描述符或接口描述符中的子类
Prot	设备描述符或接口描述符中的协议

示例

```
Mode=00000004 VID=1230 PID=1230 class=03 #输入设备在捕获模式下运行
Mode=00000002 VID=1230 PID=1230 class=03 #输入设备在交互模式下运行（默认）
Mode=00000001 VID=1230 PID=1230 class=03 #输入设备在未进行任何优化的情况下运行
Mode=00000100 VID=1230 PID=1230 # 设备设置优化禁用（默认）
Mode=00000200 VID=1230 PID=1230 # 设备设置优化启用
```

使用注册表设置禁用优化模式

可以通过注册表标记在系统范围内禁用优化模式：

```
HKLM\System\CurrentControlSet\Services\Icausb\Parameters
```

DisableInputOptimization DWORD - 将值设置为 **1**

需要重新启动系统才能使此注册表更改生效。

为 Wacom 签名板设备禁用屏幕保护程序

对于使用 Wacom 签名板和平板电脑的情况，我们建议按如下方式禁用屏幕保护程序：

- 1. 重定向设备后安装 **Wacom-STU-Driver**。
- 2. 安装 **Wacom-STU-Display MSI** 以获取签名板控制面板的访问权限。
- 3. 转至**控制面板 > Wacom STU Display > STU4 30 或 STU530**，选择所需模式对应的选项卡。
- 4. 单击 **Change**（更改），然后在弹出 UAC 安全窗口时选择 **Yes**（是）。
- 5. 选择 **Disable slideshow**（禁用幻灯片放映），然后单击 **Apply**（应用）。

为一种签名板模式设置此设置后，此设置将应用于所有模式。

客户端 USB 设备重定向

此设置允许或阻止 USB 设备与用户设备之间往来的重定向。

默认情况下，不重定向 USB 设备。

客户端 USB 设备重定向规则

此设置指定 USB 设备重定向规则。

默认情况下，不指定任何规则。

用户插入 USB 设备时，主机设备会依次根据每条策略规则对其进行检查，直至找到匹配项。任何设备的第一个匹配项都被视为最终选择。如果第一个匹配项是一条“Allow”规则，该设备会远程连接到虚拟桌面。如果第一个匹配项是一条“Deny”规则，则该设备只能连接本地桌面。如果未找到匹配项，则使用默认规则。

策略规则的格式为 {Allow:|Deny:} 后接一组以空格分隔的 tag=value 表达式。支持以下标记：

标记名称	说明

VID 标记名称	设备描述符中的供应商 ID 说明
PID	设备描述符中的产品 ID
REL	设备描述符中的版本 ID
Class	设备描述符或接口描述符中的类
SubClass	设备描述符或接口描述符中的子类
Prot	设备描述符或接口描述符中的协议

创建新策略规则时，请注意：

- 规则不区分大小写。
- 规则末尾可以带有以 # 开头的可选注释。
- 空白注释行和纯注释行会被忽略。
- 标识必须使用匹配运算符 =（例如，VID=1230_）。
- 每条规则都必须另起新行，或包含在以分号分隔的列表中。
- 参考 USB Implementers Forum, Inc. Web 站点上提供的 USB 类代码。

管理员定义的 USB 策略规则示例：

- Allow: VID=1230 PID=0007 # ANOther Industries, ANOther Flash Drive
- Deny: Class=08 subclass=05 # Mass Storage
- 要创建一条拒绝所有 USB 设备的规则，请使用未附带任何其他标记的“DENY:”。

客户端 USB 即插即用设备重定向

此设置允许或禁止在客户端会话中使用即插即用设备，例如照相机或销售点 (POS) 设备。

默认情况下，允许即插即用设备重定向。当设置为允许时，将重定向特定用户或组的即插即用设备。当设置为禁止时，将不重定向任何设备。

视频显示策略设置

May 04, 2016

“视频显示”部分中包含用于控制从虚拟桌面发送到用户设备的图像质量的策略设置。

简单图形的首选颜色深度

仅适用于 XenApp 和 XenDesktop **7.6 FP3**。

允许将简单图形的颜色深度降低到**16 位/像素**，这可能会提高低带宽连接下的响应能力，但会略微降低图像的视觉质量。此选项仅在不使用视频编解码器压缩图形时受支持。

默认情况下，设置为 24 位/像素。

目标帧速率

此设置指定每秒从虚拟桌面发送到用户设备的最大帧数。

默认情况下，最大值为 30 帧/秒。

设置一个较高的每秒帧数值（例如 30）可改进用户体验，但需要占用更多带宽。减小每秒帧数值（例如 10）可将服务器可扩展性提高至最高水平，但用户体验将非常差。对于 CPU 速度较慢的用户设备，指定较低的值可以改善用户体验。

使用视频编解码器进行压缩

仅适用于 XenApp 和 XenDesktop **7.6 FP3**。

允许视频解码在端点上可用时使用视频编解码器压缩图形。视频解码在端点上不可用时，或者当您指定了不使用视频编解码器时，将同时使用静态图像压缩和位图缓存。

默认情况下，设置为“使用视频编解码器(可用时)”。

视觉质量

此设置指定在用户设备上显示的图像所需的视觉质量。

默认情况下，此参数设置为“中”。

要指定图像质量，请选择下列选项之一：

- 低
- 中 - 在大多数用例中可提供最佳性能和最高带宽效率
- 高 - 如果需要视觉上无损的图像质量，建议采用此设置
- 设为无损 - 在高网络活动期间向用户设备发送有损图像，在网络活动减少后发送无损图像。此设置可改进带宽有限的网络连接条件下的性能
- 始终无损 - 如果保留图像数据非常重要（例如，显示不允许有质量损失的 X 光图像时），选择始终无损可确保不会将有损数据发送到用户设备。

如果在策略中启用了旧图形模式设置，视觉质量设置在此策略中无效。

移动图像策略设置

May 04, 2016

“移动图像”部分包含使您能够删除或更改动态图像的压缩的设置。

最低图像质量

注意：对于 Virtual Delivery Agent 7.x，仅当启用旧图形模式策略设置时，才会应用此策略设置。

此设置指定自适应显示功能最低可接受的图像质量。使用的压缩程度越低，所显示图像的质量越高。可以从“超高”、“很高”、“高”、“正常”或“低”压缩程度中进行选择。

默认设置为“正常”。

移动图像压缩

此设置指定是否启用自适应显示功能。自适应显示功能将根据可用带宽自动调整视频和幻灯片播放时过渡性幻灯片的图像质量。启用自适应显示功能后，用户应看到顺畅的展示效果，质量没有任何损失。

默认情况下，启用“自适应显示”功能。

对于版本 7.0 至 7.6 的 VDA，此设置仅在启用旧图像模式时应用。对于版本 7.6 FP1、FP2 和 FP3 的 VDA，此设置在启用旧图形模式时应用，或在禁用旧图形模式并且未使用视频编解码器来压缩图像的时候应用。

启用旧图形模式时，必须重新启动会话，策略更改才能生效。自适应显示与渐进式显示相互排斥；启用自适应显示将禁用渐进式显示，反之亦然。但是，可以同时禁用自适应显示和渐进式显示。渐进式显示是一项旧功能，建议不要用于 XenApp 和 XenDesktop。设置渐进式阈值级别将禁用自适应显示。

渐进式压缩级别

注意：对于 Virtual Delivery Agent 7.x，仅当启用旧图形模式策略设置时，才会应用此策略设置。

此设置提供的图像的初始显示细节较少但速度更快。

默认情况下，不应用任何渐进式压缩。

细节更丰富的图像（由常规有损压缩设置定义）在可用时显示。使用“很高”或“超高”压缩可改善需要占用大量带宽的图形（例如照片）的查看速度。

要使渐进式压缩生效，其压缩级别必须高于有损压缩级别设置。

注意：提高与渐进式压缩相关联的压缩级别，还会提高客户端连接上动态图像的交互性。动态图像（例如旋转的三维模型）的质量在图像停止动作前会暂时降低，之后会应用标准有损压缩设置。

下列策略设置为相关设置：

- 渐进式压缩阈值
- 渐进式超级压缩

渐进式压缩阈值

注意：对于 Virtual Delivery Agent 7.x，仅当启用旧图形模式策略设置时，才会应用此策略设置。

此设置表示应用渐进式压缩的连接的最大带宽 (Kbps)。此设置仅适用于低于此带宽的客户端连接。

默认情况下，该阈值为 2147483647 Kbps。

下列策略设置为相关设置：

- 渐进式压缩阈值

- 渐进式超级压缩

目标最低帧速率

此设置为动态图像指定了低带宽条件下，系统尝试保持的最低每秒帧速率。
默认情况下设置为 10 fps。

对于版本 7.0 至 7.6 的 VDA，此设置仅在启用旧图像模式时应用。对于版本 7.6 FP1 至 FP3 的 VDA，此设置在禁用或启用旧图像模式时均可以应用。

静态图像策略设置

May 04, 2016

“静态图像”部分包含使您能够删除或更改静态图像的压缩的设置。

额外颜色压缩

当通过带宽受限的客户端连接交付图像时，此设置可以允许或禁止这些图像使用额外颜色压缩，从而以降低显示图像质量的方式来提高响应能力。

默认情况下，禁用额外颜色压缩。

启用后，则只有当客户端连接带宽低于额外颜色压缩阈值的值时，才会应用额外颜色压缩。如果客户端连接带宽高于该阈值，或者选择了已禁用，则不会应用额外颜色压缩。

额外颜色压缩阈值

注意：对于 Virtual Delivery Agent 7.x，仅当启用旧图形模式策略设置时，才会应用此策略设置。

此设置表示连接的最大带宽 (Kbps)，如果低于该带宽，则会应用额外颜色压缩。如果客户端连接带宽低于设置的值，则会应用额外颜色压缩（如果启用）。

默认情况下，该阈值为 8192 Kbps。

超级压缩

注意：对于 Virtual Delivery Agent 7.x，仅当启用旧图形模式策略设置时，才会应用此策略设置。

此设置允许或禁止使用一种更为高级但会占用更多 CPU 资源的图形算法，在不损失图像质量的情况下降低渐进式压缩之外的压缩占用的带宽。

默认情况下，超级压缩功能处于禁用状态。

如果启用超级压缩，它会应用到所有有损压缩设置。这种压缩在 Citrix Receiver 上受支持，但对其他插件不起作用。

下列策略设置为相关设置：

- 渐进式压缩级别
- 渐进式压缩阈值

有损压缩级别

注意：对于 Virtual Delivery Agent 7.x，仅当启用旧图形模式策略设置时，才会应用此策略设置。

此设置控制通过带宽受限的客户端连接交付的图像上所用的有损压缩程度。在此类情况下，显示未压缩的图像速度会很慢。

默认情况下，选择中等压缩。

为改善带宽密集型图像的响应速度，请使用高压压缩。当保留图像数据非常重要（例如显示不允许有质量损失的 X 光图像时），您可能不希望使用有损压缩。

相关策略设置：有损压缩阈值

有损压缩阈值

注意：对于 Virtual Delivery Agent 7.x，仅当启用旧图形模式策略设置时，才会应用此策略设置。

此设置表示应用有损压缩的连接的最大带宽 (Kbps)。

默认情况下，该阈值为 2147483647 Kbps。

将有损压缩级别设置添加到策略且不指定阈值可以提高通过 LAN 传输的高清晰位图（例如照片）的显示速度。

相关策略设置：有损压缩级别

WebSocket 策略设置

May 04, 2016

WebSocket 部分包含用于使用 Receiver for HTML5 访问虚拟桌面和托管应用程序的策略设置。WebSocket 功能通过在基于浏览器的应用程序和服务器之间进行双向通信（无需打开多个 HTTP 连接），从而提高安全性并减少开销。

WebSocket 连接

此设置允许或禁止 WebSocket 连接。

默认情况下，禁止 WebSocket 连接。

WebSocket 端口号

此设置可识别传入的 WebSocket 连接的端口。

默认情况下，此值为 8008。

WebSocket 可信原始服务器列表

此设置提供了一个逗号分隔的可信原始服务器列表，通常是 Receiver for Web，表示为 URL。服务器仅接受来自下列地址之一的 WebSocket 连接。

默认情况下，通配符 * 用于信任所有 Receiver for Web URL。

如果您选择在列表中键入地址，请使用以下语法：

<protocol>://<Fully qualified domain name of host>[:port]

此协议应为 HTTP 或 HTTPS。如果未指定端口号，则端口 80 用于 HTTP，端口 443 用于 HTTPS。

通配符 * 可用于 URL，但不能作为 IP 地址 (10.105.*.*) 的一部分。

负载管理策略设置

May 04, 2016

“负载管理”部分包含用于在交付 Windows 服务器操作系统计算机的服务器之间启用和配置负载管理的策略设置。

并发登录容错

此设置指定服务器可以接受的最大并发登录数。

默认情况下，此参数设置为 2。

CPU 使用率

此设置指定服务器报告满负载时的 CPU 使用率，以百分比表示。启用时，服务器报告满负载的默认值是 90%。

默认情况下，此设置处于禁用状态，负载计算中不包括 CPU 使用率。

排除 CPU 使用率的进程优先级

此设置指定从 CPU 使用率负载指数中排除进程 CPU 使用率时的优先级。

默认情况下，此参数设置为“低于正常”或“低”。

磁盘使用情况

此设置指定服务器报告 75% 满负载时的磁盘队列长度。启用时，磁盘队列长度的默认值为 8。

默认情况下，此设置处于禁用状态，负载计算中不包括磁盘使用情况。

最大会话数

此设置指定服务器可以托管的最大会话数。启用时，服务器可托管最大会话数的默认设置为 250。

默认情况下，此设置处于启用状态。

内存使用率

此设置指定服务器报告满负载时的内存使用率，以百分比表示。启用时，服务器报告满负载的默认值是 90%。

默认情况下，此设置处于禁用状态，负载计算中不包括内存使用率。

内存使用基础负载

此设置指定基本操作系统内存使用空间的近似值，并定义服务器被视为零负载时的内存使用空间（以 MB 为单位）。

默认情况下，将其设置为 768 MB。

Profile Management 策略设置

Nov 02, 2016

The Profile Management section contains policy settings for enabling profile management and specifying which groups to include in and exclude from profile management processing.

Other information (such as the names of the equivalent .ini file settings and which version of profile management is required for a policy setting) is available in [Profile Management Policies](#).

高级策略设置

May 04, 2016

“高级设置”部分包含与 Profile Management 高级配置相关的策略设置。

禁用自动配置

此设置允许 Profile Management 检查您的环境；例如，检查是否存在个人虚拟磁盘，并相应地配置组策略。系统仅调整处于“未配置”状态的 Profile Management 策略，因此之前所做的任何自定义设置都将保留。此功能加快了部署，并简化了优化过程。此功能无需任何配置，但您可以在升级（保留早期版本的设置）或进行故障排除时禁用自动配置。自动配置在 XenApp 或其他环境中无法使用。

默认情况下允许自动配置。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果在此处和 .ini 文件中均未配置此设置，将开启自动配置功能，以便当环境发生变化时可以更改 Profile Management 设置。

遇到问题时注销用户

此设置允许 Profile Management 在遇到问题（例如，用户存储不可用）时注销用户。启用时，在用户被注销前，系统会向用户显示一条错误消息。禁用时，系统会为用户提供一个临时配置文件。

默认情况下，此设置处于禁用状态，遇到问题时，系统会为用户提供一个临时配置文件。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果在此处和 .ini 文件中均未配置此设置，将提供一个临时配置文件。

访问锁定文件的重试次数

此设置指定 Profile Management 尝试访问锁定文件的次数。

默认情况下，此参数设置为重试 5 次。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果在此处和 .ini 文件中均未配置此设置，将使用默认值。

注销时处理 Internet Cookie 文件

此设置允许 Profile Management 在注销时处理 index.dat，从而删除持续浏览后保留在文件系统中可能会导致配置文件膨胀的 Internet Cookie。启用此设置会延长注销时间，因此，请仅在遇到此问题时才启用此设置。

默认情况下，此设置处于禁用状态，Profile Management 不会在注销时处理 index.dat。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也在 .ini 文件中，则将不处理 Index.dat。

基本策略设置

May 04, 2016

“基本设置”部分包含与 Profile Management 基本配置有关的策略设置。

活动写回

此设置允许在会话过程中或注销之前将修改后的文件和文件夹（但不包括注册表设置）同步到用户存储。

默认情况下，禁止在会话过程中同步到用户存储。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 .ini 文件中，则将处于启用状态。

启用 Profile Management

此设置允许 Profile Management 处理登录和注销。

默认情况下，此设置已禁用，以方便部署。

重要：Citrix 建议仅在执行所有其他设置任务并测试 Citrix 用户配置文件在环境中的执行情况后，再启用 Profile Management。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则 Profile Management 不会以任何方式处理 Windows 用户配置文件。

排除的组

此设置指定从 Profile Management 处理范围中排除哪些计算机本地组和域组（本地组、全局组和通用组）排除之外。

启用时，Profile Management 不会处理指定用户组的成员。

默认情况下，此设置处于禁用状态，此时将处理所有用户组的成员。

按 <域名>\<组名> 格式指定域组。

如果未在此处配置此设置，将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则会处理所有用户组的成员。

脱机配置文件支持

此设置可启用脱机配置文件支持，使配置文件能够在网络连接断开后尽早与用户存储进行同步。

默认情况下，脱机配置文件支持处于禁用状态。

此设置适用于使用便携式计算机或移动设备的漫游用户。网络连接断开时，即使在便携式计算机或移动设备重新启动或进入休眠状态后，设备上的配置文件仍将保持原样。移动用户工作时，其配置文件将在本地更新，并在重新建立网络连接时与用户存储进行同步。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果在此处和 .ini 文件中均未配置此设置，将禁用对脱机配置文件的支持。

用户存储路径

此设置指定用于保存用户设置（如注册表设置和同步文件）的目录路径（用户存储）。

默认情况下，将使用主驱动器上的 Windows 目录。

如果禁用此设置，用户设置将保存在主目录的 Windows 子目录中。

路径可以是：

- **相对路径**。此路径必须是相对于主目录的路径，通常配置为 Active Directory 中用户的 #homeDirectory# 属性。
- **绝对 UNC 路径**。此路径通常指定服务器共享或 DFS 命名空间。
- **已禁用或未配置**。在此情况下，假设值为 #homeDirectory#\Windows。

配置此策略设置时，请使用以下类型的变量：

- 百分号括起的系统环境变量（例如 %ProfVer%）。请注意，系统环境变量通常需要额外设置。
- 井号括起的 Active Directory 用户对象属性（例如 #sAMAccountName#）。
- Profile Management 变量。有关详细信息，请参阅 Profile Management 文档。

您也可以使用 %username% 和 %userdomain% 用户环境变量并创建自定义属性，以完全定义位置或用户等组织变量。属性区分大小写。

示例：

- \\server\share\#sAMAccountName# 将用户设置存储到 UNC 路径 \\server\share\JohnSmith（如果当前用户的 #sAMAccountName# 解析为 JohnSmith）
- \\server\profiles\$\%USERNAME%.%USERDOMAIN%\!CTX_PROFILEVER!!CTX_OSBITNESS!可以扩展为 \\server\profiles\$\JohnSmith.DOMAINCONTROLLER1\v2x64

重要：无论使用哪种属性或变量，均请确认此设置是否可以扩展到包含 NTUSER.DAT 的文件夹的上层文件夹。例如，如果此文件包含在 \\server\profiles\$\JohnSmith.Finance\v2x64\UPM_Profile 中，则应将用户存储路径设置为 \\server\profiles\$\JohnSmith.Finance\v2x64，而非 \UPM_Profile 子文件夹。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果未在此处也未在 .ini 文件中配置此设置，则将使用主驱动器上的 Windows 目录。

处理本地管理员登录

此设置指定是否处理 BUILTIN\Administrators 组成员的登录。这使具有本地管理员权限的域用户（通常是已分配虚拟桌面的用户）可以跳过任何处理过程而直接登录，并对出现 Profile Management 问题的桌面进行故障排除。

如果在服务器操作系统上禁用或未配置该设置，Profile Management 会假定必须处理域用户登录，但不处理本地管理员登录。在桌面操作系统上，会处理本地管理员登录。

默认情况下，此设置处于禁用状态，不会处理本地管理员登录。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果在此处和 .ini 文件中均未配置此设置，则不会处理本地管理员登录。

处理的组

此设置指定哪些计算机本地组和域组（本地组、全局组和通用组）包括在 Profile Management 处理范围内。

启用时，Profile Management 仅处理指定用户组的成员。

默认情况下，此设置处于禁用状态，此时将处理所有用户组的成员。

按 <域名>\<组名> 格式指定域组。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则会处理所有用户组的成员。

跨平台策略设置

May 04, 2016

“跨平台”部分包含与配置 Profile Management 跨平台设置功能有关的策略设置。

跨平台设置用户组

此设置指定在跨平台设置功能启用时处理其配置文件的 Windows 用户组。

默认情况下，此设置处于禁用状态，系统会处理在处理的组策略设置中指定的所有用户组。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 .ini 文件中，则会处理所有用户组。

启用跨平台设置

此设置启用或禁用跨平台设置功能，允许在用户连接到多个操作系统中运行的同一应用程序时迁移和漫游用户的配置文件。

默认情况下，跨平台设置功能处于禁用状态。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果未在此处也未在 .ini 文件中配置此设置，则将不应用任何跨平台设置。

跨平台定义路径

此设置将从下载包复制的定义文件的网络位置指定为 UNC 路径。

注意：用户必须对此位置具有读取权限，管理员必须对此位置具有写入权限，并且此位置必须是服务器消息块 (SMB) 或通用 Internet 文件系统 (CIFS) 文件共享。

默认情况下，不指定路径。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果未在此处也未在 .ini 文件中配置此设置，则将不应用任何跨平台设置。

跨平台设置存储路径

此设置指定跨平台设置存储的路径，即用于保存用户跨平台设置的文件夹。该路径可以是 UNC 路径，也可以是相对于主目录的路径。

注意：用户必须对跨平台设置存储具有写入权限。

默认情况下，此设置处于禁用状态，此时将使用路径 Windows\PM_CP。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果在此处和 .ini 文件中均未配置此设置，将使用默认值。

创建跨平台设置的来源

如果在平台的 OU 中启用此设置，则此设置会将某个平台指定为基础平台。数据将从基础平台的配置文件迁移到跨平台设置存储中。

每个平台自有的一组配置文件存储在独立的 OU 中。这意味着您必须决定使用哪个平台的配置文件数据来设置跨平台设置存储。这称为基础平台。

启用时，如果跨平台设置存储中包含无任何数据的定义文件，或者单平台配置文件中的缓存数据比存储中的定义数据新，则 Profile Management 会将数据从单平台配置文件迁移到存储中。

重要：如果在多个 OU、多个用户或计算机对象中启用了此设置，则第一位用户登录到的平台将作为基础配置文件。默认情况下，此设置处于禁用状态，Profile Management 不会将数据从单平台配置文件迁移到存储中。

文件系统策略设置

May 04, 2016

“文件系统”部分包含的策略设置用于配置用户配置文件中的哪些文件和目录将在安装配置文件的系统和用户存储之间进行同步。

排除策略设置

May 04, 2016

“排除”部分包含的策略设置用于配置将用户配置文件中的哪些文件和目录从同步过程中排除。

排除列表 - 目录

此设置指定用户配置文件中将在同步过程中忽略的文件夹的列表。

以相对于用户配置文件的路径 (%USERPROFILE%) 指定文件夹名。

默认情况下，此设置处于禁用状态，系统会同步用户配置文件中的所有文件夹。

示例：Desktop 忽略用户配置文件中的 Desktop 文件夹

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果在此处和 .ini 文件中均未配置此设置，则将对用户配置文件中的所有文件夹进行同步。

排除列表 - 文件

此设置指定用户配置文件中将在同步过程中忽略的文件的列表。

默认情况下，此设置处于禁用状态，系统会同步用户配置文件中的所有文件。

以相对于用户配置文件的路径 (%USERPROFILE%) 指定文件名。请注意，允许使用通配符，但以递归方式应用。

示例：Desktop\Desktop.ini 忽略 Desktop 文件夹中的 Desktop.ini 文件

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果在此处和 .ini 文件中均未配置此设置，则将对用户配置文件中的所有文件进行同步。

同步策略设置

May 04, 2016

“同步”部分包含的策略设置用于指定将在安装配置文件的系统与用户存储之间同步用户配置文件中的哪些文件和文件夹。

同步的目录

此设置指定您希望 Profile Management 在同步过程中包含的已排除文件夹中的任何文件。默认情况下，Profile Management 将同步用户配置文件中的所有内容。无需通过将用户配置文件的子文件夹添加到该列表中来包括这些子文件夹。有关详细信息，请参阅[包含和排除项目](#)。

该列表中的路径必须是相对于用户配置文件的路径。

示例：Desktop\exclude\include 确保同步 include 子文件夹，即使不同步 Desktop\exclude 文件夹

默认情况下，此设置处于禁用状态，不指定任何文件夹。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则仅对用户配置文件中没有排除的文件夹进行同步。

同步的文件

此设置指定您希望 Profile Management 在同步过程中包含的已排除文件夹中的任何文件。默认情况下，Profile Management 将同步用户配置文件中的所有内容。无需通过将用户配置文件中的文件添加到该列表中来包括这些文件。有关详细信息，请参阅[包含和排除项目](#)。

该列表中的路径必须是相对于用户配置文件的路径。相对路径解释为相对于用户配置文件。允许使用通配符，但只能用于文件名。不能嵌套通配符，将递归应用通配符。

示例：

- AppData\Local\Microsoft\Office\Access.qat 指定了默认配置中排除的文件夹中的文件
- AppData\Local\MyApp*.cfg 指定了配置文件夹 AppData\Local\MyApp 及其子文件夹中扩展名为 .cfg 的所有文件

默认情况下，此设置处于禁用状态，不指定任何文件。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则仅会对用户配置文件中没有排除的文件夹进行同步。

镜像的文件夹

此设置指定要镜像的、相对于用户配置文件根文件夹的文件夹。配置此策略设置有助于解决与任何事务性文件夹（也称为引用文件夹）有关的问题，该文件夹包含互相依赖的文件，即其中一个文件引用其他文件。

通过镜像文件夹，Profile Management 可将事务性文件夹及其内容作为单个实体进行处理，从而避免配置文件膨胀。请注意，在这些情况下，“后写入内容有效”。因此，镜像的文件夹中包含的在多个会话中被修改的文件会被最新的更新覆盖，导致配置文件更改丢失。

例如，您可以镜像 Internet Explorer Cookie 文件夹，从而可以将 Index.dat 与其索引的 Cookie 同步。

假设用户具有两个 Internet Explorer 会话，分别位于不同的服务器上，并且服务器在每个会话中访问不同的站点，则每个站点

的 Cookie 会添加到相应的服务器。用户从第一个会话注销时（或者在会话过程中，前提是配置了活动写回功能），第二个会话中的 Cookie 应替代第一个会话中的 Cookie。但是，这两个会话却合并在一起，而且对 Index.dat 中的 Cookie 的引用将过期。进一步浏览新会话会导致重复合并以及 Cookie 文件夹膨胀。

镜像 Cookie 文件夹可解决上述问题，因为该操作在每次用户注销时都将用最后一次会话中的 Cookie 覆盖之前的 Cookie，从而使 Index.dat 保持为最新。

默认情况下，此设置处于禁用状态，不镜像任何文件夹。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果未在此处也未在 .ini 文件中配置此策略，则不会镜像任何文件夹。

文件夹重定向策略设置

May 04, 2016

“文件夹重定向”部分包含的策略设置用于指定是否将经常出现在配置文件中的文件夹重定向到共享网络位置。

授予管理员访问权限

此设置使管理员可以访问用户重定向的文件夹的内容。

默认情况下，此设置处于禁用状态，用户被授予独占访问其重定向文件夹内容的权限。

包含域名

此设置允许将 %userdomain% 环境变量包含在为重定向文件夹指定的 UNC 路径中。

默认情况下，此设置处于禁用状态，%userdomain% 环境变量不包括在为重定向文件夹指定的 UNC 路径中。

“AppData (漫游)”策略设置

May 04, 2016

“AppData (漫游)”部分包含的策略设置用于指定是否将“AppData (漫游)”文件夹的内容重定向到共享网络位置。

“AppData (漫游)”路径

此设置指定“AppData (漫游)”文件夹的内容将重定向到的网络位置。

默认情况下，此设置处于禁用状态，不指定任何位置。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

“AppData (漫游)”的重定向设置

此设置指定如何重定向“AppData (漫游)”文件夹的内容。

默认情况下，内容重定向到 UNC 路径。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

“联系人”策略设置

May 04, 2016

“联系人”部分包含的策略设置用于指定是否将“联系人”文件夹的内容重定向到共享网络位置。

“联系人”路径

此设置指定“联系人”文件夹的内容将重定向到的网络位置。

默认情况下，此设置处于禁用状态，不指定任何位置。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

“联系人”的重定向设置

此设置指定如何重定向“联系人”文件夹的内容。

默认情况下，内容重定向到 UNC 路径。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

桌面策略设置

May 04, 2016

“桌面”部分包含的策略设置用于指定是否将“桌面”文件夹的内容重定向到共享网络位置。

“桌面”路径

此设置指定“桌面”文件夹的内容将重定向到的网络位置。

默认情况下，此设置处于禁用状态，不指定任何位置。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

“桌面”的重定向设置

此设置指定如何重定向“桌面”文件夹的内容。

默认情况下，内容重定向到 UNC 路径。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

“文档”策略设置

May 04, 2016

“文档”部分包含的策略设置用于指定是否将“文档”文件夹的内容重定向到共享网络位置。

“文档”路径

此设置指定“文档”文件夹中的文件将重定向到的网络位置。

默认情况下，此设置处于禁用状态，不指定任何位置。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

必须启用文档路径设置，以将文件重定向到“文档”文件夹，同时将文件重定向到“音乐”、“图形”和“视频”文件夹。

“文档”的重定向设置

此设置指定如何重定向“文档”文件夹的内容。

默认情况下，内容重定向到 UNC 路径。

要控制如何重定向“文档”文件夹的内容，请选择以下选项之一：

- 重定向到以下 UNC 路径。将内容重定向到“文档”路径策略设置中指定的 UNC 路径。
- 重定向到用户的主目录。将内容重定向到用户主目录，通常配置为 Active Directory 中用户的 #homeDirectory# 属性。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

“下载”策略设置

May 04, 2016

“下载”部分包含的策略设置用于指定是否将“下载”文件夹的内容重定向到共享网络位置。

“下载”路径

此设置指定“下载”文件夹中的文件将重定向到的网络位置。

默认情况下，此设置处于禁用状态，不指定任何位置。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

“下载”的重定向设置

此设置指定如何重定向“下载”文件夹的内容。

默认情况下，内容重定向到 UNC 路径。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

“收藏夹”策略设置

May 04, 2016

“收藏夹”部分包含的策略设置用于指定是否将“收藏夹”文件夹的内容重定向到共享网络位置。

“收藏夹”路径

此设置指定“收藏夹”文件夹的内容将重定向到的网络位置。

默认情况下，此设置处于禁用状态，不指定任何位置。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

“收藏夹”的重定向设置

此设置指定如何重定向“收藏夹”文件夹。

默认情况下，内容重定向到 UNC 路径。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

“链接”策略设置

May 04, 2016

“链接”部分包含的策略设置用于指定是否将“链接”文件夹的内容重定向到共享网络位置。

“链接”路径

此设置指定“链接”文件夹的内容将重定向到的网络位置。

默认情况下，此设置处于禁用状态，不指定任何位置。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

“链接”的重定向设置

此设置指定如何重定向“链接”文件夹的内容。

默认情况下，内容重定向到 UNC 路径。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

“音乐”策略设置

May 04, 2016

“音乐”部分包含的策略设置用于指定是否将“音乐”文件夹的内容重定向到共享网络位置。

“音乐”路径

此设置指定“音乐”文件夹的内容将重定向到的网络位置。

默认情况下，此设置处于禁用状态，不指定任何位置。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

“音乐”的重定向设置

此设置指定如何重定向“音乐”文件夹的内容。

默认情况下，内容重定向到 UNC 路径。

要控制如何重定向“音乐”文件夹的内容，请选择以下选项之一：

- 重定向到以下 UNC 路径。将内容重定向到“音乐”路径策略设置中指定的 UNC 路径。
- 重定向到相对于“文档”文件夹的路径。将内容重定向到相对于“文档”文件夹的文件夹。

要将内容重定向到相对于“文档”文件夹的文件夹，必须启用“文档”路径设置。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

“图片”策略设置

May 04, 2016

“图片”部分包含的策略设置用于指定是否将“图片”文件夹的内容重定向到共享网络位置。

“图片”路径

此设置指定“图片”文件夹的内容将重定向到的网络位置。

默认情况下，此设置处于禁用状态，不指定任何位置。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

“图片”的重定向设置

此设置指定如何重定向“图片”文件夹的内容。

默认情况下，内容重定向到 UNC 路径。

要控制如何重定向“图片”文件夹的内容，请选择以下选项之一：

- 重定向到以下 UNC 路径。将内容重定向到“图片”路径策略设置中指定的 UNC 路径。
- 重定向到相对于“文档”文件夹的路径。将内容重定向到相对于“文档”文件夹的文件夹。

要将内容重定向到相对于“文档”文件夹的文件夹，必须启用“文档”路径设置。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

“保存的游戏”策略设置

May 04, 2016

“保存的游戏”部分包含的策略设置用于指定是否将“保存的游戏”文件夹的内容重定向到共享网络位置。

此设置指定如何重定向“保存的游戏”文件夹的内容。

默认情况下，内容重定向到 UNC 路径。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

此设置指定“保存的游戏”文件夹的内容将重定向到的网络位置。

默认情况下，此设置处于禁用状态，不指定任何位置。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

“搜索”策略设置

May 04, 2016

“搜索”部分包含的策略设置用于指定是否将“搜索”文件夹的内容重定向到共享网络位置。

此设置指定如何重定向“搜索”文件夹的内容。

默认情况下，内容重定向到 UNC 路径。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

此设置指定“搜索”文件夹的内容将重定向到的网络位置。

默认情况下，此设置处于禁用状态，不指定任何位置。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

“开始菜单”策略设置

May 04, 2016

“开始菜单”部分包含的策略设置用于指定是否将“开始菜单”文件夹的内容重定向到共享网络位置。

此设置指定如何重定向“开始菜单”文件夹的内容。

默认情况下，内容重定向到 UNC 路径。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

此设置指定“开始菜单”文件夹的内容将重定向到的网络位置。

默认情况下，此设置处于禁用状态，不指定任何位置。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

“视频”策略设置

May 04, 2016

“视频”部分包含的策略设置用于指定是否将“视频”文件夹的内容重定向到共享网络位置。

此设置指定如何重定向“视频”文件夹的内容。

默认情况下，内容重定向到 UNC 路径。

要控制如何重定向“视频”文件夹的内容，请选择以下选项之一：

- 重定向到以下 UNC 路径。将内容重定向到“视频”路径策略设置中指定的 UNC 路径。
- 重定向到相对于“文档”文件夹的路径。将内容重定向到相对于“文档”文件夹的文件夹。

要将内容重定向到相对于“文档”文件夹的文件夹，必须启用“文档”路径设置。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

此设置指定“视频”文件夹的内容将重定向到的网络位置。

默认情况下，此设置处于禁用状态，不指定任何位置。

如果未在此处配置此设置，Profile Management 将不会重定向指定文件夹。

“日志”策略设置

May 04, 2016

“日志”部分包含的策略设置用于配置 Profile Management 日志记录。

此设置启用或禁用对 Active Directory 中执行的操作进行详细日志记录。

默认情况下，禁用此设置。

启用此设置时，请确保启用日志记录设置也处于启用状态。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则记录错误和常规信息。

此设置启用或禁用常规信息的详细日志记录。

默认情况下，禁用此设置。

启用此设置时，请确保启用日志记录设置也处于启用状态。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则记录错误和常规信息。

此设置启用或禁用常见警告的详细日志记录。

默认情况下，禁用此设置。

启用此设置时，请确保启用日志记录设置也处于启用状态。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则记录错误和常规信息。

此设置启用或禁用调试（详细日志记录）模式下的 Profile Management 日志记录。在调试模式中，大量的状态信息记录在“%SystemRoot%\System32\Logfiles\UserProfileManager”下的日志文件中。

默认情况下，此设置处于禁用状态，只记录错误。

Citrix 建议您仅在对 Profile Management 进行故障排除时才启用此设置。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则只记录错误。

此设置启用或禁用对文件系统中执行的操作进行详细日志记录。

默认情况下，禁用此设置。

启用此设置时，请确保启用日志记录设置也处于启用状态。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则记录错误和常规信息。

此设置启用或禁用文件系统通知的详细日志记录。

默认情况下，禁用此设置。

启用此设置时，请确保启用日志记录设置也处于启用状态。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则记录错误和常规信息。

此设置启用或禁用用户注销的详细日志记录。

默认情况下，禁用此设置。

启用此设置时，请确保启用日志记录设置也处于启用状态。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则记录错误和常规信息。

此设置启用或禁用用户登录的详细日志记录。

默认情况下，禁用此设置。

启用此设置时，请确保启用日志记录设置也处于启用状态。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则记录错误和常规信息。

此设置指定 Profile Management 日志文件的最大允许大小（字节）。

默认情况下，设置为 1048576 字节 (1 MB)。

如果您有足够的磁盘空间，Citrix 建议您将此文件的大小增加到 5 MB 或更高。如果日志文件大小超出最大大小，则将删除现有文件备份 (.bak)，将日志文件重命名为 .bak，并创建一个新日志文件。

日志文件在 %SystemRoot%\System32\Logfiles\UserProfileManager 中创建。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果在此处和 .ini 文件中均未配置此设置，将使用默认值。

此设置指定用于保存 Profile Management 日志文件的备用路径。

默认情况下，此设置处于禁用状态，日志文件保存在默认位置：%SystemRoot%\System32\Logfiles\UserProfileManager。

该路径可以指向本地驱动器或基于网络的远程驱动器（UNC 路径）。远程路径在大型分布式环境中非常有用，但可能会带来大量网络流量，因此可能不适用于日志文件。对于已置备的具有静态硬盘驱动器的虚拟机，应设置该驱动器的一个本地路径。这样可以确保重新启动虚拟机时能够保留日志文件。对于没有静态硬盘驱动器的虚拟机，设置一个 UNC 路径将使您能够保留日志文件，但该虚拟机的系统帐户必须具有 UNC 共享的写入权限。对于受脱机配置文件功能管理的任何便携式计算机，应使用本地路径。

如果对日志文件使用的是 UNC 路径，则 Citrix 建议对日志文件文件夹应用恰当的访问控制列表，以确保只有授权用户或计算机帐户能够访问存储的文件。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则使用默认位置 %SystemRoot%\System32\Logfiles\UserProfileManager。

此设置启用或禁用个性化用户信息的详细日志记录。

默认情况下，禁用此设置。

启用此设置时，请确保启用日志记录设置也处于启用状态。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则记录错误和常规信息。

此设置启用或禁用用户登录及注销时策略值的详细日志记录。

默认情况下，禁用此设置。

启用此设置时，请确保启用日志记录设置也处于启用状态。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则记录错误和常规信息。

此设置启用或禁用在注册表中执行的操作的详细日志记录。

默认情况下，禁用此设置。

启用此设置时，请确保启用日志记录设置也处于启用状态。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则记录错误和常规信息。

此设置启用或禁用用户注销时任何注册表差异的详细日志记录。

默认情况下，禁用此设置。

启用此设置时，请确保启用日志记录设置也处于启用状态。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则记录错误和常规信息。

“配置文件处理”策略设置

May 04, 2016

“配置文件处理”部分包含的策略设置用于配置 Profile Management 对用户配置文件的处理方式。

此设置指定注销时 Profile Management 在删除本地缓存的配置文件之前的可选延迟时间（分钟）。

值为 0 时会在注销过程结束时立即删除配置文件。Profile Management 每分钟检查一次注销，因此值为 60 可确保在用户注销后一到两分钟内删除配置文件（取决于最后一次检查的时间）。如果已知注销期间进程会使文件或用户注册表配置单元处于打开状态，延长延迟时间将很有用。对于大型配置文件，这种做法还可以加快注销速度。

默认情况下，此参数设置为 0，Profile Management 会立即删除本地缓存的配置文件。

启用此设置时，请确保注销时删除本地缓存的配置文件也处于启用状态。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果在此处和 .ini 文件中均未配置此设置，则会立即删除配置文件。

此设置指定在用户注销后是否删除本地缓存的配置文件。

如果启用此设置，用户注销后，将删除其本地配置文件缓存。Citrix 建议您为终端服务器启用此设置。

默认情况下，此设置处于禁用状态，用户注销后，将继续保留用户本地配置文件缓存。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则不会删除缓存的配置文件。

在既存在用户存储中的用户配置文件，又存在本地 Windows 用户配置文件（非 Citrix 用户配置文件）的情况下，此设置用于配置 Profile Management 的行为。

默认情况下，Profile Management 将使用本地 Windows 配置文件，但不通过任何方式更改该配置文件。

要控制 Profile Management 的行为，请选择以下选项之一：

- 使用本地配置文件。Profile Management 将使用本地配置文件，但不通过任何方式更改该配置文件。
- 删除本地配置文件。Profile Management 将删除本地 Windows 用户配置文件，然后导入用户存储中的 Citrix 用户配置文件。
- 重命名本地配置文件。Profile Management 将重命名本地 Windows 用户配置文件（用于备份），然后导入用户存储中的 Citrix 用户配置文件。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则使用现有本地配置文件。

此设置指定当用户在用户存储中没有当前配置文件时，会在登录期间迁移到用户存储的配置文件的类型。

如果用户在用户存储中没有配置文件，则登录期间 Profile Management 可以即时迁移现有配置文件。此后，Profile Management 将在当前会话以及通过相同用户存储路径配置的任何其他会话中使用用户存储配置文件。

默认情况下，会在登录期间将本地配置文件和漫游配置文件迁移到用户存储。

要指定登录期间迁移到用户存储的配置文件的类型，请选择以下选项之一：

- 本地配置文件和漫游配置文件
- 本地
- 漫游
- 无（已禁用）

如果选择无，系统将使用现有 Windows 机制创建新配置文件，就像在未安装 Profile Management 的环境中一样。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 .ini 文件中，则将迁移现有的本地配置文件和漫游配置文件。

此设置指定希望 Profile Management 用来创建新用户配置文件的模板配置文件的路径。

指定的路径必须为文件夹的完整路径，其中包含 NTUSER.DAT 注册表文件以及模板配置文件所需的其他任何其他文件夹和文件。

注意：请勿在路径中包括 NTUSER.DAT。例如，对于文件 \\myservername\myprofiles\template\ntuser.dat，应将路径设置为 \\myservername\myprofiles\template。

应使用绝对路径，绝对路径可以是 UNC 路径，也可以是本地计算机上的路径。例如，可以使用后者指定永久存在于 Citrix Provisioning Services 映像中的模板配置文件。不支持相对路径。

注意：此设置不支持扩展 Active Directory 属性、系统环境变量或 %USERNAME% 和 %USERDOMAIN% 变量。

默认情况下，此设置处于禁用状态，系统将根据用户首次登录的设备上的默认用户配置文件创建新用户配置文件。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则不会使用任何模板。

此设置允许在创建新用户配置文件时以模板配置文件覆盖本地配置文件。

如果用户没有 Citrix 用户配置文件，但存在本地 Windows 用户配置文件，默认情况下将使用本地配置文件（如果没有禁用迁移，还将会迁移至用户存储）。启用此策略设置后，模板配置文件可以覆盖在创建新用户配置文件时所使用的本地配置文件。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则不会使用任何模板。

此设置可在创建新用户配置文件时以模板配置文件覆盖漫游配置文件。

如果用户没有 Citrix 用户配置文件，但存在漫游 Windows 用户配置文件，则默认情况下将使用漫游配置文件（如果没有禁用迁移，还会迁移至用户存储）。启用此策略设置后，模板配置文件可以覆盖在创建新用户配置文件时所使用的漫游配置文件。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则不会使用任何模板。

此设置使 Profile Management 可以将模板配置文件用作创建所有新用户配置文件时使用的默认配置文件。

默认情况下，此设置处于禁用状态，系统将根据用户首次登录的设备上的默认用户配置文件创建新用户配置文件。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果该设置没有在此配置，也不在 INI 文件中，则不会使用任何模板。

“注册表”策略设置

May 04, 2016

“注册表”部分包含的策略设置用于指定在 Profile Management 处理中要包含或排除的注册表项。

此设置指定 HKCU 注册表配置单元中注册表项的列表，当用户注销时，这些注册表项将从 Profile Management 处理中排除。

如果启用，当用户注销时，会从处理中排除此列表中指定的注册表项。

默认情况下，此设置处于禁用状态，当用户注销时，会处理 HKCU 注册表配置单元中的所有注册表项。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果在此处和 .ini 文件中均未配置此设置，则不会从处理中排除任何注册表项。

此设置指定 HKCU 注册表配置单元中注册表项的列表，当用户注销时，这些注册表项将包括在 Profile Management 处理中。

如果启用，当用户注销时，仅处理此列表中指定的注册表项。

默认情况下，此设置处于禁用状态，当用户注销时，会处理 HKCU 注册表配置单元中的所有注册表项。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果在此处和 .ini 文件中均未配置此设置，将处理整个 HKCU。

“流用户配置文件”策略设置

May 04, 2016

“流用户配置文件”部分包含的策略设置用于指定 Profile Management 处理流用户配置文件的方式。

此设置指定 Profile Management 在用户登录后是否立即缓存流文件。在用户登录后缓存文件可以节省网络带宽，增强用户体验。

将此设置与 Profile Streaming 设置结合使用。

默认情况下，此设置处于禁用状态，用户登录后不会立即缓存流文件。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果未在此处也未在 .ini 文件中配置此设置，则将处于禁用状态。

此设置指定通过流技术推送的文件大小的下限 (MB)。Profile Management 会在用户登录后立即缓存任何等于或大于此大小的文件。

默认情况下，将其设置为 0（零），并使用缓存整个配置文件功能。启用缓存整个配置文件功能时，Profile Management 会在用户登录后，通过后台任务提取用户存储中的所有配置文件内容。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果未在此处也未在 .ini 文件中配置此设置，则将处于禁用状态。

此设置启用和禁用 Citrix 流用户配置文件功能。如果启用，只有当用户在登录后访问配置文件中的文件和文件夹时，这些文件和文件夹才会从用户存储提取到本地计算机中。注册表项以及挂起区域中的文件会立即提取。

默认情况下，Profile Streaming 处于禁用状态。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果未在此处也未在 .ini 文件中配置此设置，则将处于禁用状态。

此设置基于 Windows 用户组指定通过流技术推送 OU 中的哪些用户配置文件。

启用时，仅通过流技术推送指定用户组中的用户配置文件。所有其他用户配置文件将按正常方式进行处理。

默认情况下，此设置处于禁用状态，OU 中的所有用户配置文件将按正常方式进行处理。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果在此处和 .ini 文件中均未配置此设置，则会处理所有用户配置文件。

此设置指定一个一天为单位的时间段，如果服务器无响应并且用户存储处于锁定状态，则经过这一时间段后，用户文件从挂起区域写回到用户存储。这样可以防止挂起区域膨胀，并保证用户存储始终包含最新的文件。

默认情况下，此参数设置为 1（一）天。

如果未在此处配置此设置，则将使用 .ini 文件中的值。

如果在此处和 .ini 文件中均未配置此设置，将使用默认值。

Receiver 策略设置

May 04, 2016

Receiver 部分包含的策略设置用于指定要推送到虚拟桌面上运行的 Receiver for Windows 的 Storefront 地址列表。

此设置指定一个 StoreFront 存储列表，管理员可选择将其推送到在虚拟桌面上运行的 Receiver for Windows。创建交付组时，管理员可选择要将哪些存储推送到在此组内的虚拟桌面上运行的 Receiver for Windows。

默认情况下，不指定任何存储。

对于每个存储，以分号分隔条目的形式指定以下信息：

- 应用商店名称。向存储用户显示的名称。
- 存储 URL。存储的 URL。
- 存储启用状态。存储是否可供用户使用。此状态为“启用”或“禁用”。
- 存储说明。显示给存储用户的说明。

例如：Sales Store;https://sales.mycompany.com/Citrix/Store/discovery;On;Store for Sales staff

Virtual Delivery Agent 策略设置

May 04, 2016

Virtual Delivery Agent (VDA) 部分包含的策略设置可以控制 VDA 与站点控制器之间的通信。

重要：如果没有使用自动更新功能，VDA 需要使用这些设置提供的信息向 Delivery Controller 注册。由于此信息是进行注册所必需的信息，因此，除非在 VDA 安装期间提供了此信息，否则必须使用组策略编辑器配置下列设置：

- 控制器注册 IPv6 网络掩码
- 控制器注册端口
- 控制器 SID
- 控制器
- 仅使用 IPv6 控制器注册
- 站点 GUID

此策略设置允许管理员将 VDA 限制为仅在首选的子网（而非全局 IP，如果已注册）中使用。此设置指定 VDA 要注册的 IPv6 地址和网络。VDA 将仅在与指定网络掩码匹配的第二个地址上进行注册。仅当启用仅使用 IPv6 控制器注册策略设置时，此设置才有效。

默认情况下，此设置为空。

仅当禁用启用控制器自动更新设置时，才使用此设置。

此设置指定 VDA 向控制器注册时所用的 TCP/IP 端口号（如果使用基于注册表的注册方式）。

默认情况下，此端口号设置为 80。

仅当禁用启用控制器自动更新设置时，才使用此设置。

此设置指定 VDA 向控制器注册时所用的控制器安全标识符 (SID) 的空格分隔列表（如果使用基于注册表的注册方式）。此设置为可选设置，可与控制器设置结合使用，用以限制可用于注册的控制器的列表。

默认情况下，此设置为空。

仅当禁用启用控制器自动更新设置时，才使用此设置。

此设置用于指定 VDA 向控制器注册时所用的控制器完全限定域名 (FQDN) 的空格分隔列表（如果使用基于注册表的注册方式）。此设置为可选设置，可与控制器 SID 设置结合使用。

默认情况下，此设置为空。

通过此设置，VDA 可在安装后自动向控制器注册。

VDA 注册后，注册的控制器的列表将向 VDA 发送当前控制器的 FQDN 和 SID 的列表。VDA 会将此列表写入到静态存储。每个控制器还

将每隔 90 分钟在站点数据库中检查一次控制器信息；如果在上次检查后添加或删除了控制器，或者如果策略发生更改，控制器将向注册的 VDA 发送更新后的列表。VDA 将接受所接收最新列表中的所有控制器的连接。

默认情况下，此设置处于启用状态。

此设置可控制 VDA 向控制器注册时所用的地址格式：

- 启用后，VDA 将使用计算机的 IPv6 地址向控制器注册。当 VDA 与控制器进行通信时，将使用以下地址顺序：全局 IP 地址、唯一本地地址 (ULA)、链接本地地址（如果没有其他可用的 IPv6 地址）。
- 禁用后，VDA 将使用计算机的 IPv4 地址向控制器注册并与之通信。

默认情况下，禁用此设置。

仅当禁用启用控制器自动更新设置时，才使用此设置。

此设置用于指定 VDA 向控制器注册时所用的站点的全局唯一标识符 (GUID)（如果使用基于 Active Directory 的注册方式）。

默认情况下，此设置为空。

HDX 3D Pro 策略设置

May 04, 2016

HDX 3D Pro 部分包含用于为用户启用和配置图像质量配置工具的策略设置。该工具使用户能够实时调整图像质量与响应速度之间的平衡点，从而优化对可用带宽的使用情况。

此设置指定用户是否能够使用图像质量配置工具启用和禁用无损压缩功能。默认情况下，用户可以选择启用无损压缩功能。

用户启用无损压缩功能后，图像质量将自动设置为在图像配置工具中可用的最大值。默认情况下，可以根据用户设备和主机计算机的功能，使用基于 GPU 或 CPU 的压缩。

此设置指定图像质量配置工具中用于定义用户可用的图像质量调整范围的最小值和最大值。

可以指定 0 到 100 之间（包括 0 和 100）的图像质量值。最大值必须大于或等于最小值。

虚拟 IP 策略设置

May 04, 2016

“虚拟 IP”部分包含的策略设置用于控制会话是否具有自己的虚拟环回地址。

启用此设置时，每个会话具有自己的虚拟环回地址。禁用时，会话不具有单独的虚拟环回地址。

默认情况下，禁用此设置。

此设置指定可使用虚拟环回地址的应用程序可执行文件。将程序添加到列表时，仅指定可执行文件名称，无需指定完整路径。

默认情况下，不指定任何可执行文件。

使用注册表配置 COM 端口和 LPT 端口重定向设置

May 04, 2016

用于 COM 端口和 LPT 端口重定向的策略设置位于 VDA 映像或计算机上的 HKLM\Software\Citrix\GroupPolicy\Defaults\Deprecated 下方。

要启用 COM 端口和 LPT 端口重定向，请添加类型为 REG_DWORD 的新注册表项，如下所示：

警告：注册表编辑不当会导致严重问题，可能导致需要重新安装操作系统。Citrix 无法保证因“注册表编辑器”使用不当导致出现的问题能够得以解决。使用“注册表编辑器”需自担风险。在编辑注册表之前，请务必进行备份。

注册表项	说明	允许使用的值
AllowComPortRedirection	允许或禁止使用 COM 端口重定向	1（允许）或 0（禁止）
LimitComBw	COM 端口重定向通道的带宽限制	数值
LimitComBWPercent	COM 端口重定向通道的带宽限制（占总会话带宽的百分比）	0 到 100 之间的数值
AutoConnectClientComPorts	从用户设备自动连接 COM 端口	1（允许）或 0（禁止）
AllowLptPortRedirection	允许或禁止使用 LPT 端口重定向	1（允许）或 0（禁止）
LimitLptBw	LPT 端口重定向通道的带宽限制	数值
LimitLptBwPercent	LPT 端口重定向通道的带宽限制（占总会话带宽的百分比）	0 到 100 之间的数值
AutoConnectClientLptPorts	从用户设备自动连接 LPT 端口	1（允许）或 0（禁止）

配置这些设置后，请修改您的计算机目录，使其使用新主映像或更新的物理机。用户下次注销时，将使用新设置更新桌面。

Connector for Configuration Manager 2012 策略设置

May 04, 2016

Connector for Configuration Manager 2012 部分包含用于配置 Citrix Connector 7.5 代理的策略设置。

重要：警告、注销和重新启动消息策略仅适用于手动管理或由 Provisioning Services 管理的服务器操作系统计算机目录的部署。对于这些计算机目录，当存在待定的应用程序安装或软件更新时，Connector 服务将向用户发出警报。对于 MCS 管理的目录，使用 Studio 通知用户。对于手动管理的桌面操作系统目录，使用 Configuration Manager 通知用户。对于由 Provisioning Services 管理的桌面操作系统目录，使用 Provisioning Services 通知用户。

此设置定义将提前警告消息显示给用户的时间间隔。

间隔使用 ddd.hh:mm:ss 格式进行设置，其中：

- ddd 代表天数，是介于 0 到 999 之间的可选参数。
- hh 为介于 0 到 23 之间的小时数。
- mm 为介于 0 到 59 之间的分钟数。
- ss 是介于 0 到 59 之间的秒数。

默认情况下，间隔设置为 1 小时 (01:00:00)。

此设置包含显示给用户的可编辑消息文本，用以通知用户即将进行软件更新或维护，需要用户注销。

默认情况下，消息为：“{TIMESTAMP} Please save your work.The server will go offline for maintenance in {TIMELEFT}” ({TIMESTAMP} 请保存您的工作。服务器将在 {TIMELEFT} 内脱机进行维护)

此设置包含显示给用户的提前警告消息的可编辑标题栏文本。

默认情况下，标题为：“Upcoming Maintenance” (即将进行维护)

此设置定义在维护前多久首次显示提前警告消息。

时间使用 ddd.hh:mm:ss 格式进行设置，其中：

- ddd 代表天数，是介于 0 到 999 之间的可选参数。
- hh 为介于 0 到 23 之间的小时数。
- mm 为介于 0 到 59 之间的分钟数。
- ss 是介于 0 到 59 之间的秒数。

默认情况下，该设置为 16 小时 (16:00:00)，表示第一个提前警告消息大约在维护前 16 小时显示。

此设置包含可编辑的消息文本，警告用户开始强制注销。

默认情况下，消息为：“The server is currently going offline for maintenance” (服务器当前即将脱机进行维护)

此设置包含最终强制注销消息的可编辑标题栏文本。

默认情况下，标题为：“Notification From IT Staff”（来自 IT 人员的通知）

此设置定义从通知用户注销到实施强制注销以处理待解决维护之间的时间段。

时间使用 ddd.hh:mm:ss 格式进行设置，其中：

- ddd 代表天数，是介于 0 到 999 之间的可选参数。
- hh 为介于 0 到 23 之间的小时数。
- mm 为介于 0 到 59 之间的分钟数。
- ss 是介于 0 到 59 之间的秒数。

默认情况下，强制注销宽限期设置为 5 分钟 (00:05:00)。

此设置包含可编辑的文本，在开始强制注销之前通知用户保存其工作并注销。

默认情况下，此消息中包含以下内容：“{TIMESTAMP} Please save your work and log off.The server will go offline for maintenance in {TIMELEFT}”（{TIMESTAMP} 请保存您的工作。服务器将在 {TIMELEFT} 内脱机进行维护）

此设置包含强制注销消息的标题栏的可编辑文本。

默认情况下，标题为：“Notification From IT Staff”（来自 IT 人员的通知）

Connector Agent 将自动检测其是在 Provisioning Services 还是 MCS 管理的计算机克隆上运行。该 Agent 会阻止 Configuration Manager 在托管映像克隆上更新，并自动在目录的主映像上安装更新。

更新主映像后，请使用 Studio 调配 MCS 目录克隆的重新启动行为。在 Configuration Manager 维护时段，Connector Agent 将自动调配 PVS 目录克隆的重新启动行为。要覆盖此行为以便软件由 Configuration Manager 安装在目录克隆上，请将托管映像模式更改为已禁用。

此设置包含可编辑的消息文本，用于在服务器即将重新启动时通知用户。

默认情况下，消息为：“The server is currently going offline for maintenance”（服务器当前即将脱机进行维护）

此设置决定 Citrix Connector Agent 任务的运行频率。

时间使用 ddd.hh:mm:ss 格式进行设置，其中：

- ddd 代表天数，是介于 0 到 999 之间的可选参数。
- hh 为介于 0 到 23 之间的小时数。
- mm 为介于 0 到 59 之间的分钟数。

- ss 是介于 0 到 59 之间的秒数。

默认情况下，常规时间间隔设置为 5 分钟 (00:05:00)。

打印

May 04, 2016

在您的环境下管理打印机的过程分为多个阶段：

1. 熟悉打印概念（如果您还不熟悉）。
2. 规划打印体系结构。此阶段包括分析业务需求、现有打印基础结构、用户和应用程序当前与打印过程的交互方式，以及哪种打印管理模式最适合您的环境。
3. 选择打印机置备方法，然后创建部署打印设计的策略，以配置打印环境。在添加新员工或服务器时更新策略。
4. 为用户部署打印配置前，首先对试验配置进行测试。
5. 管理打印机驱动程序并优化打印性能，以维护 Citrix 打印环境。
6. 对可能发生的问题进行故障排除。

在开始规划部署之前，一定要了解有关打印的以下核心概念：

- 可用的打印机置备类型
- 如何路由打印作业
- 打印机驱动程序管理基础知识

打印概念建立在 Windows 打印概念的基础上。要在您的环境下配置并成功管理打印，您必须了解 Windows 网络和客户端打印的工作原理，及其在此环境下的相应打印行为。

在此环境下，所有打印都在托管应用程序的计算机上由用户启动。打印作业通过网络打印服务器或用户设备重定向到打印设备。

虚拟桌面和应用程序的用户没有永久工作区。会话结束后，用户的工作区将被删除，因此在每个会话开始时需要重新构建所有设置。这样，每次用户启动新会话时，系统都必须重新构建用户的工作区。

用户执行打印时：

- 确定向用户提供的打印机。此过程也称作打印机置备。
- 恢复用户的打印首选项。
- 确定会话的默认打印机。

您可以通过配置打印机置备、打印作业路由、打印机属性保留以及驱动程序管理等选项来自定义这些任务的执行方式。请务必评估各种选项设置对您环境中的打印性能及用户体验有何影响。

在会话中启用打印机的过程称为置备。打印机置备通常采用动态处理方式，即不会预先确定和存储会话中出现的打印机，而是在登录和重新连接期间建立会话时基于策略来装配打印机。因此，打印机会随着策略、用户位置以及网络变化（只要策略中反映了这些内容）而变化。这样，漫游到不同位置的用户可以看到其工作区的变化。

系统还会监视客户端打印机，并根据客户端打印机的添加、删除和更改情况动态调整在会话中自动创建的打印机。动态打印机发现对移动用户很有益，因为他们从各种设备进行连接。

最常用的打印机置备方法有：

- **通用打印服务器** - Citrix [通用打印服务器](#) 为网络打印机提供通用打印支持。通用打印服务器使用通用打印驱动程序。通过此

解决方案，您可以使用服务器操作系统计算机上的单个驱动程序以允许从任何设备进行网络打印。

Citrix 建议针对远程打印服务器的情况使用 Citrix 通用打印服务器。通用打印服务器通过网络以经过优化和压缩的格式传输打印作业，从而最大程度地减少网络使用，并改善用户体验。

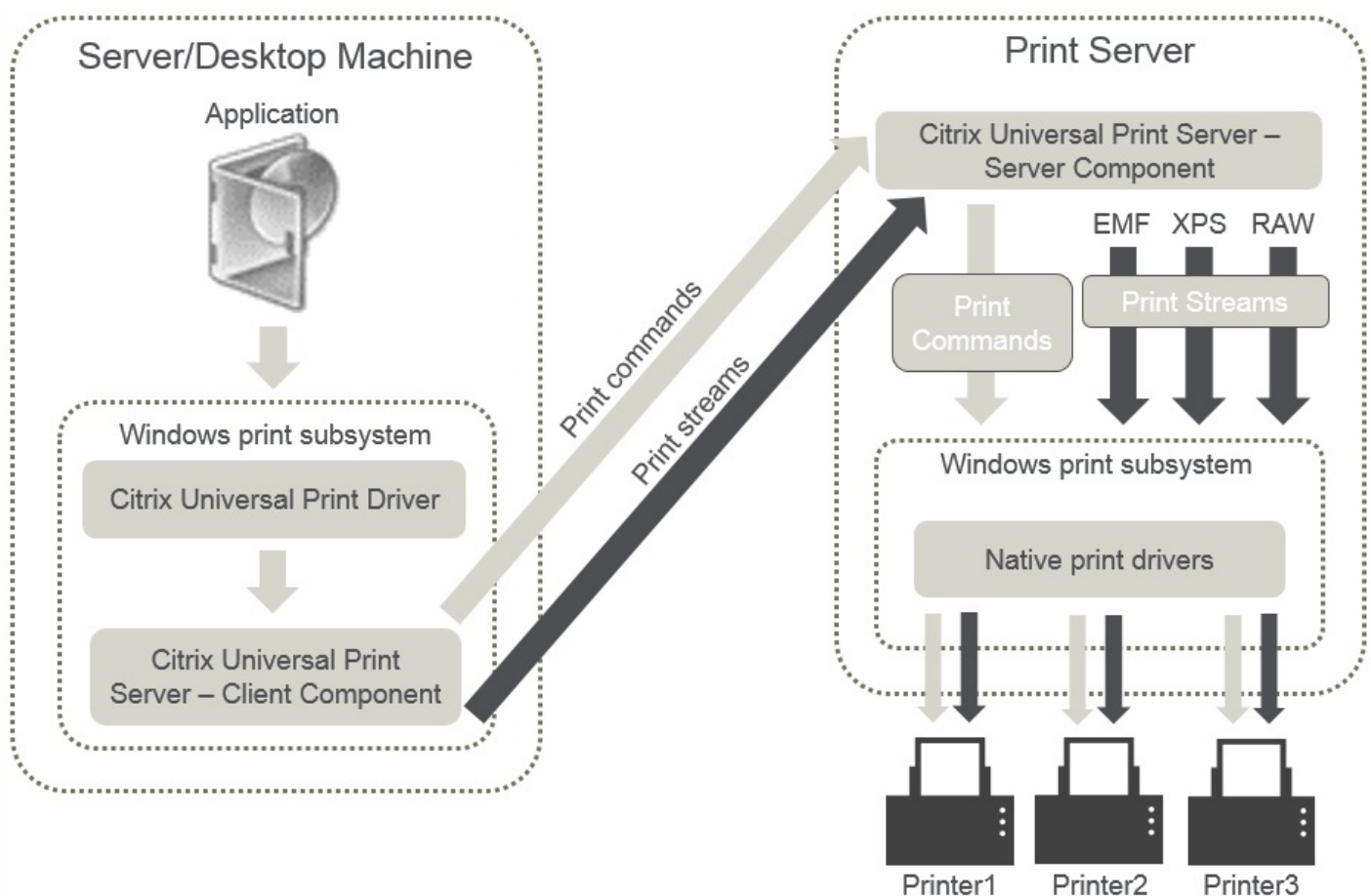
通用打印服务器功能包含以下组件：

- 客户端组件（通用打印客户端）- 在置备会话网络打印机并且使用通用打印驱动程序的每台服务器操作系统计算机上启用通用打印客户端。
- 服务器组件（通用打印服务器）- 在置备会话网络打印机并且对会话打印机使用通用打印驱动程序的每台打印服务器上安装通用打印服务器（不管会话打印机是否集中置备）。

有关通用打印服务器的要求和设置的详细信息，请参阅系统要求和安装文档。

注意：VDI-in-a-Box 5.3 同样支持通用打印服务器。有关利用 VDI-in-a-Box 安装通用打印服务器的信息，请参阅 VDI-in-a-Box 文档。

下图显示了在使用通用打印服务器的环境中基于网络的打印机的典型工作流。



启用 Citrix 通用打印服务器时，所有连接的网络打印机都会通过自动发现利用该服务器。

- **自动创建** - 自动创建指每次启动会话时自动创建的打印机。远程网络打印机和本地连接的客户端打印机都可自动创建。对每个用户具有大量打印机的环境，请考虑仅自动创建默认客户端打印机。自动创建的打印机数量越少，服务器操作系统计算机需要的开销（内存和 CPU）就越少。尽量减少自动创建的打印机数量还可以缩短用户登录时间。

自动创建的打印机基于：

- 用户设备上安装的打印机。
- 适用于会话的任何策略。

通过自动创建策略，您可以限制自动创建的打印机的数量或类型。默认情况下，在用户设备上自动配置所有打印机（包括本地连接的打印机和网络打印机）时，打印机会在会话中启用。

用户结束会话后，该会话使用的打印机将被删除。

客户端和网络打印机自动创建的维护工作彼此关联。例如，要添加打印机，需要执行以下操作：

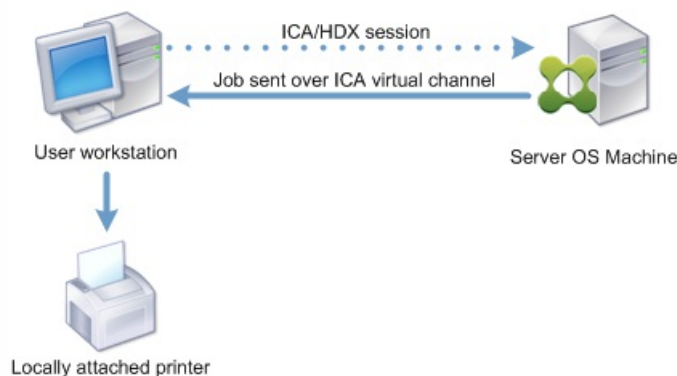
- 更新会话打印机策略设置。
- 使用“打印机驱动程序映射和兼容性”策略设置向所有服务器操作系统计算机添加驱动程序。

术语打印途径涉及两个方面：路由打印作业的路径以及对打印作业进行后台打印的位置。此概念的这两方面都很重要。路由会影响网络流量。后台打印会影响对处理打印作业的设备上的本地资源的使用。

在此环境中，打印作业可以由两种途径传送到打印设备：通过客户端或通过网络打印服务器。这两种途径称为客户端打印途径和网络打印途径。默认情况下选择哪种路径取决于所使用的打印机类型。

本地连接的打印机

系统将作业从服务器操作系统计算机通过客户端路由到本地连接的打印机，然后再路由到打印设备。ICA 协议将优化和压缩打印作业流量。打印设备本地连接到用户设备时，打印作业将通过 ICA 虚拟通道进行路由。



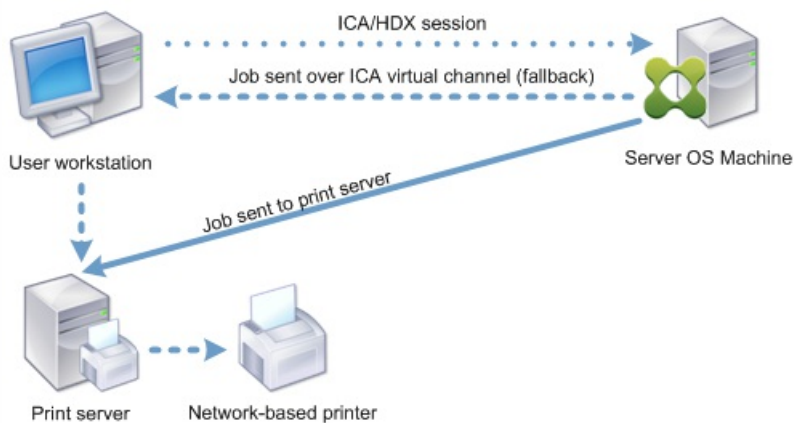
基于网络的打印机

默认情况下，发往网络打印机的所有打印作业都会从服务器操作系统计算机通过网络直接路由到打印服务器。但是在以下情形中，打印作业会自动通过 ICA 连接进行路由：

- 如果虚拟桌面或应用程序无法连接打印服务器。
- 如果本机打印机驱动程序在服务器操作系统计算机上不可用。

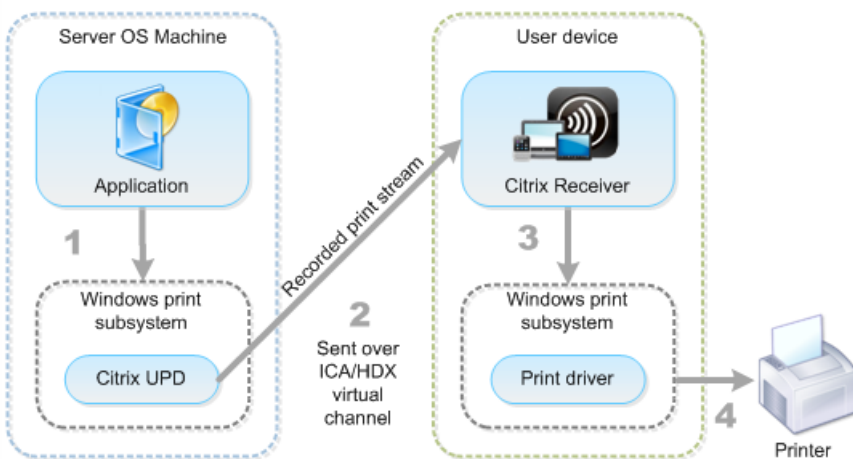
如果未启用通用打印服务器，配置面向网络打印的客户端打印途径对低带宽连接（例如广域网）非常有用，这是因为通过 ICA 连接发送作业时会进行流量优化和压缩。

此外，客户端打印途径还允许您限制流量或限制分配给打印作业的带宽。如果不能通过用户设备路由作业，例如对于没有打印功能的瘦客户端，应将服务质量配置为优先处理 ICA/HDX 流量，并确保用户在会话中获得良好的体验。



为简化此环境中的打印，Citrix 建议使用 Citrix 通用打印驱动程序。通用打印驱动程序是一种独立于设备的驱动程序，支持任何打印设备，它使得所需驱动程序的数量得以减少，从而简化了管理。

下图显示了通用打印驱动程序组件和本地连接到设备的打印机的典型工作流。

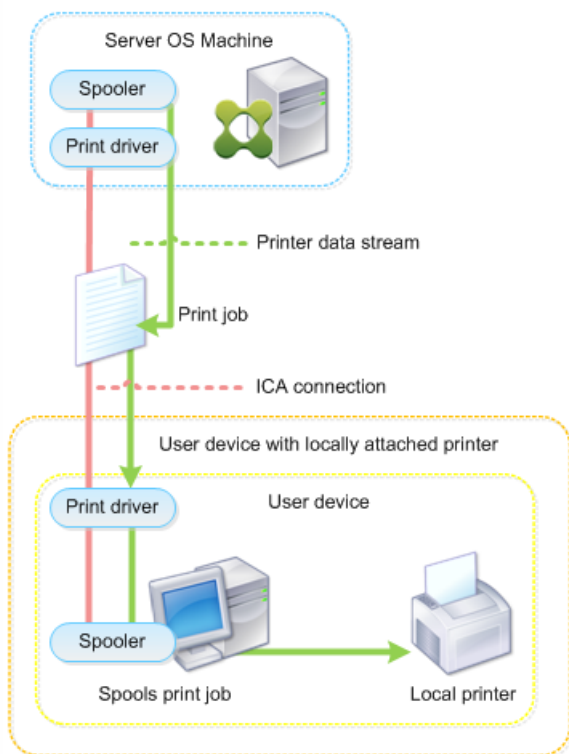


规划驱动程序管理策略时，请确定支持的驱动程序类型：通用打印驱动程序、设备特定的驱动程序或者两者。如果支持标准驱动程序，您需要确定：

- 要支持的驱动程序类型。
- 当服务器操作系统计算机中缺少打印机驱动程序时，是否要自动安装打印机驱动程序。
- 是否要创建驱动程序兼容性列表。

在自动创建打印机期间，如果系统检测到有新的本地打印机连接至用户设备，即会在服务器操作系统计算机中检查是否有所需的打印机驱动程序。默认情况下，如果 Windows 本机驱动程序不可用，系统将使用通用打印驱动程序。

要使打印成功，服务器操作系统计算机上的打印机驱动程序和用户设备上的驱动程序必须匹配。下图显示了如何在两个位置使用打印机驱动程序进行客户端打印。



- 打印配置示例
- 最佳做法、安全注意事项和默认操作
- 打印策略和首选项
- 置备打印机
- 维护打印环境
- 通用打印服务器的要求

打印配置示例

May 04, 2016

根据您的需求和环境选择最合适的打印配置方案可以简化管理工作。尽管默认打印配置使用户可以在大多数环境中进行打印，但默认设置可能无法在您的环境中提供预期的用户体验或最佳网络使用率和管理开销。

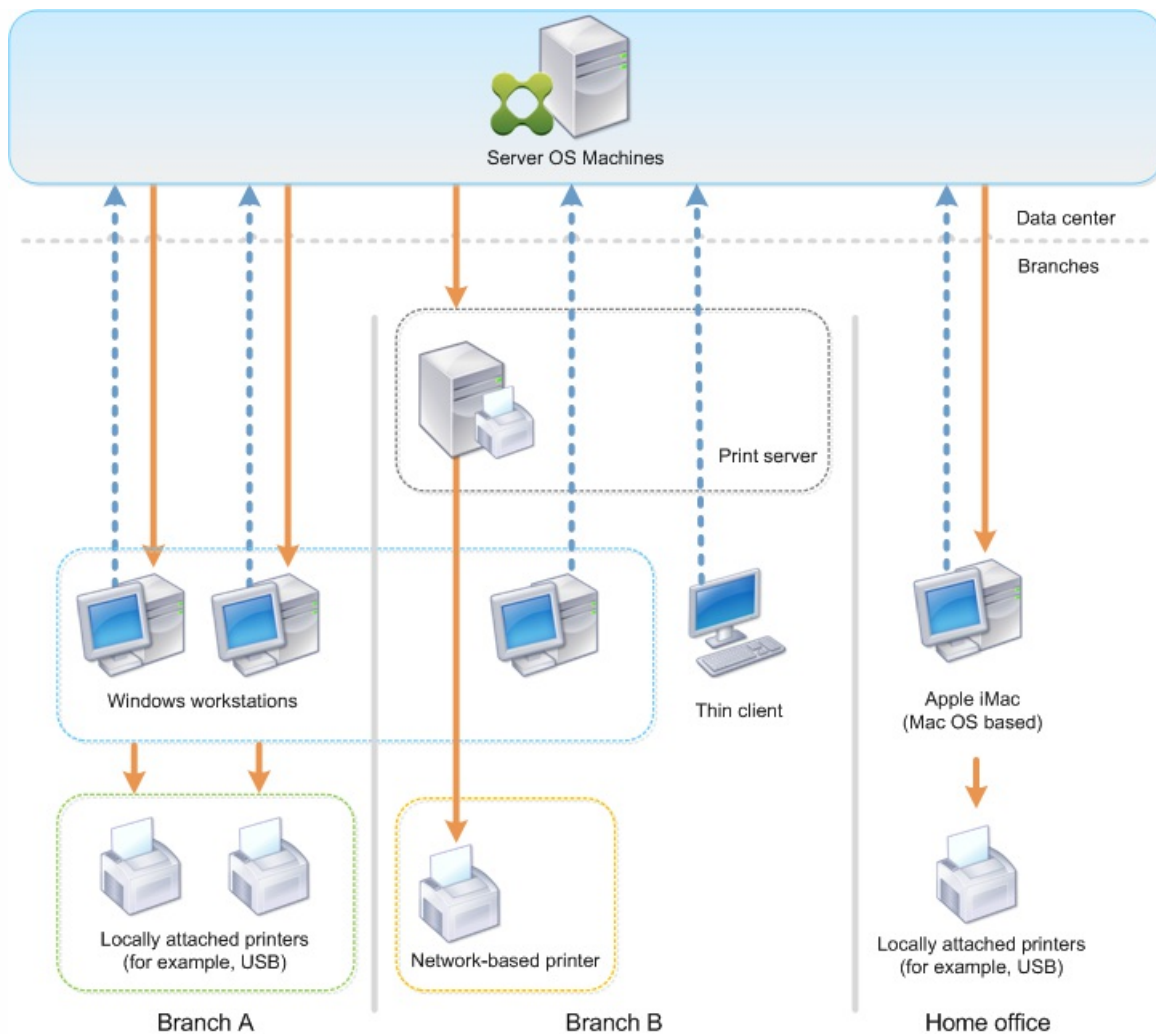
打印配置取决于：

- 业务需求以及现有的打印基础设施。
应根据您公司的需求来设计打印配置。定义打印配置时，现有的打印实现（用户是否可以添加打印机、哪些用户对哪些打印机拥有访问权限等）可以作为非常有用的参考。
- 组织是否设置了为特定用户保留专用打印机（例如人力资源或薪资专用打印机）的安全策略。
- 用户离开主要工作场所时是否需要打印，例如在不同工作站之间移动办公或者出差的工作人员。

在设计打印配置时，应努力为会话中的用户提供与从本地用户设备打印时相同的体验。

下图显示了这些用例的打印部署：

- **分支机构 A** – 小型海外分支机构，具有少数几个 Windows 工作站。每个用户工作站都有一个本地连接的专用打印机。
- **分支机构 B** – 大型分支机构，具有瘦客户端和基于 Windows 的工作站。为了提高效率，此分支机构的用户共享基于网络的打印机（每个楼层一台）。位于分支机构内部的基于 Windows 的打印服务器管理着打印队列。
- **公司总部** – 公司总部，具有基于 Mac 操作系统的用户设备，可访问公司的 Citrix 基础结构。用户设备具有本地连接的打印机。



以下部分介绍了可最大程度地降低环境复杂性并简化其管理的配置。

在分支机构 A 中，所有用户在基于 Windows 的工作站上工作，因此将使用自动创建的客户端打印机和通用打印机驱动程序。这些技术具有以下优势：

- 性能 – 打印作业通过 ICA 打印通道交付，这样可以压缩打印数据，从而节省带宽。为了确保打印大型文档的单个用户不会降低其他用户的会话性能，配置了一个 Citrix 策略以指定最大打印带宽。

备选解决方案为利用多流 ICA 连接，在此连接中，打印流量在单独的低优先级 TCP 连接中进行传输。多流 ICA 适用于不在 WAN 连接上实施服务质量 (QoS) 时使用。

- 灵活性 – 使用 Citrix 通用打印机驱动程序，可确保还可以从虚拟桌面或应用程序会话使用连接到客户端的所有打印机，而无需在数据中心的集成新打印机驱动程序。

在分支机构 B 中，所有打印机均基于网络并在 Windows 打印服务器上管理其队列，这样 Citrix 通用打印服务器便成为最有效的配置。

本地管理员在打印服务器上安装并管理所有必需的打印机驱动程序。将打印机映射到虚拟桌面或应用程序会话的工作流程如下：

- 对于基于 Windows 的工作站 – 本地 IT 团队帮助用户将基于网络的相应打印机连接到其 Windows 工作站。这样用户即可从

本地安装的应用程序进行打印。

在虚拟桌面或应用程序会话期间，本地配置的打印机通过自动创建进行枚举。然后，虚拟桌面或应用程序将作为直接网络连接连接到打印服务器（如果可能）。

将安装并启用 Citrix 通用打印服务器组件，这样就不需要使用本机打印机驱动程序。如果更新驱动程序或修改打印队列，则无需在数据中心进行任何其他配置。

- 对于瘦客户端 – 对于瘦客户端用户，必须在虚拟桌面或应用程序会话内部连接打印机。为了给用户提供最简单的打印体验，管理员为每个楼层配置了一个 Citrix 会话打印机策略，以连接各楼层的默认打印机。
为确保即使用户在楼层之间移动也能连接正确的打印机，请基于瘦客户端的子网或名称过滤策略。此配置称为邻近打印，允许维护本地打印机驱动程序（根据委派管理模式）。

如果需要修改或添加打印队列，Citrix 管理员必须修改环境中相应的会话打印机策略。

由于将在 ICA 虚拟通道外部发送网络打印流量，因此必须实施 QoS。ICA/HDX 通信使用的端口上的入站和出站网络流量优先于所有其他网络流量。该配置可确保用户会话不受大型打印作业的影响。

公司总部的用户是非标准工作站工作并使用非托管打印设备，因此最简单的方法是使用自动创建的客户端打印机和通用打印机驱动程序。

概括而言，部署示例如下所示进行配置：

- 未在服务器操作系统计算机上安装任何打印机驱动程序。仅使用 Citrix 通用打印机驱动程序。禁用回退到本机打印和自动安装打印机驱动程序。
- 将策略配置为对所有用户自动创建所有客户端打印机。默认情况下，服务器操作系统计算机将直接连接到打印服务器。所需的唯一配置是启用通用打印服务器组件。
- 对分支机构 B 的每个楼层配置会话打印机策略，并应用于相应楼层的所有瘦客户端。
- 对分支机构 B 实施 QoS，以确保卓越的用户体验。

最佳做法、安全注意事项和默认操作

May 04, 2016

多种因素决定了特定环境的最佳打印解决方案。其中一些最佳做法可能不适用于您的站点。

- 使用 Citrix 通用打印服务器。
- 使用通用打印机驱动程序或 Windows 本机驱动程序。
- 最大程度减少服务器操作系统计算机上安装的打印机驱动程序的数量。
- 使用映射到本机驱动程序的驱动程序。
- 切勿在生产站点上安装未经测试的打印机驱动程序。
- 避免更新驱动程序。而应尝试卸载驱动程序，重新启动打印服务器，然后安装替代的驱动程序。
- 卸载未使用的驱动程序或使用打印机驱动程序映射和兼容性策略，以防止通过驱动程序创建打印机。
- 尝试避免使用第 2 版内核模式驱动程序。
- 要确定打印机型号是否受支持，请联系制造商或在 www.citrix.com/ready 上查看 Citrix Ready 产品指南。

一般而言，Microsoft 提供的所有打印机驱动程序都已经过终端服务测试，保证可以与 Citrix 结合使用。但是，在使用第三方打印机驱动程序之前，请咨询打印机驱动程序供应商，以确保该驱动程序已经过 Windows Hardware Quality Labs (WHQL) 程序的终端服务认证。Citrix 不为打印机驱动程序提供认证。

Citrix 打印解决方案采用安全设计。

- Citrix Print Manager Service 会持续监视并响应会话事件，例如登录与注销、断开连接、重新连接以及会话终止。它通过模仿实际会话用户来处理服务请求。
- Citrix 打印在会话中为每台打印机分配唯一的命名空间。
- Citrix 打印为自动创建的打印机设置默认安全描述符，以确保一个会话中自动创建的客户端打印机无法被其他会话中运行的用户所访问。默认情况下，管理员用户不会意外地打印到其他会话的客户端打印机，即使他们可以看到任何客户端打印机并手动调整其权限也是如此。

默认情况下，如果未配置任何策略规则，打印行为如下所述：

- 通用打印服务器处于禁用状态。
- 在每个会话开始时自动创建在用户设备上配置的所有打印机。
此行为等效于通过自动创建所有客户端打印机选项配置 Citrix 策略设置自动创建客户端打印机。
- 系统将所有排队等候用户设备所连接的本地打印机的打印作业作为客户端打印作业进行路由（使用 ICA 通道或通过用户设备）。
- 系统将所有排队等候网络打印机的打印作业直接从服务器操作系统计算机进行路由。如果系统无法通过网络来路由打印作业，它会将这些作业作为重定向的客户端打印作业通过用户设备进行路由。
此行为等效于禁用 Citrix 策略设置直接连接到打印服务器。
- 系统会尝试将打印属性存储在用户设备上，打印属性是用户打印首选项和打印设备特定设置的组合。如果客户端不支持此操作，系统会将打印属性存储在服务器操作系统计算机上的用户配置文件中。
此行为等效于通过仅当未保存在客户端时才保留在配置文件中选项配置 Citrix 策略设置打印机属性保留。
- 系统使用 Windows 版本的打印机驱动程序（如果该驱动程序在服务器操作系统计算机上可用）。如果该打印机驱动程序不可用，系统会尝试从 Windows 操作系统中安装该驱动程序。如果 Windows 中没有提供该驱动程序，XenDesktop 将使用

Citrix 通用打印驱动程序。

此行为等效于通过仅当请求的驱动程序不可用时才使用通用打印启用 Citrix 策略设置自动安装现成的打印机驱动程序并配置通用打印设置。

启用自动安装现有的打印机驱动程序可能会导致安装大量本机打印机驱动程序。

注意：如果不确定用于打印的原始默认设置，可以通过创建新策略并将所有打印策略规则设置为“启用”来显示这些默认设置。显示的选项即为默认设置。

XenApp 和 XenDesktop 7.6 FP3 包含用于 VDA 上的打印服务器和打印子系统的始终启用日志记录功能。

为了收集 ZIP 格式的日志以便通过电子邮件发送或自动上载到 Citrix Insights Services，请使用随 7.6 FP3 中的 VDA 安装程序提供的 PowerShell cmdlet (Start-TelemetryUpload)。

打印策略和首选项

May 04, 2016

用户从已发布的应用程序访问打印机时，可以配置 Citrix 策略以指定以下设置：

- 如何设置打印机（或者如何将其添加到会话）
- 如何路由打印作业
- 如何管理打印机驱动程序

针对不同的用户设备、用户或过滤策略时所依据的任何其他对象，可以设置不同的打印配置。

大多数打印功能都是通过 Citrix 打印策略配置的。打印设置遵循标准 Citrix 策略行为。

如果用户的网络帐户有足够权限，系统可以在会话结束时将打印机设置写入打印机对象，或写入客户端打印设备。默认情况下，Receiver 在其他位置查找设置和首选项之前，将使用存储在会话中的打印机对象的设置。

默认情况下，系统在用户设备上（如果设备支持）或在服务器操作系统计算机上的用户配置文件中存储或保留打印机属性。如果用户在会话期间更改打印机属性，这些更改会在计算机上的用户配置文件中更新。下次用户登录或重新连接时，用户设备会继承这些保留的设置。即，用户必须注销并重新登录，用户设备上的打印机属性更改才会影响当前会话。

在 Windows 打印环境中，对打印首选项所做的更改可以保存在本地计算机或文档中。在此环境中，用户修改打印设置时，设置将保存在以下位置：

- **在用户设备上** – Windows 用户可以在用户设备上更改设备设置，方法是在“控制面板”中的打印机上单击鼠标右键并选择打印首选项。例如，如果选择横向作为页面方向，则将把横向保存为该打印机的默认页面方向首选项。
- **在文档内部** – 在文字处理和桌面排版程序中，页面方向等文档设置通常保存在文档中。例如，排列文档进行打印时，Microsoft Word 通常将您指定的打印首选项（例如页面方向和打印机名称）保存在文档中。下次打印该文档时，默认情况下会显示这些设置。
- **从用户在会话期间所做的更改中** – 如果在会话中通过“控制面板”进行更改（即在服务器操作系统计算机上），系统将仅保留对自动创建的打印机的打印设置所做的更改。
- **在服务器操作系统计算机上** – 这些是与计算机上特定打印机驱动程序关联的默认设置。

根据用户做出更改的位置，任何基于 Windows 的环境中保留的设置均会有所差异。也就是说，出现在一个位置（例如电子表格程序中）的打印设置会与其他位置（例如文档中）的打印设置有所差别。因此，应用到特定打印机的打印设置在整个会话过程中可能会发生变化。

由于打印首选项可以保存在多个位置，因此系统会根据特定优先级对其进行处理。此外，必须注意的是，设备设置与文档设置相互独立且通常优先于文档设置。

默认情况下，系统始终优先应用用户在会话期间修改的打印设置（即保留的设置），然后才会考虑其他设置。当用户打印时，系统会将存储在服务器操作系统计算机上的默认打印机设置与任何保留的设置或客户端打印机设置进行合并然后应用。

Citrix 建议您不要更改打印机属性的存储位置。默认设置为将打印机属性保存在用户设备上，这是确保打印属性一致的最简便方法。如果系统无法在用户设备上保存属性，则会自动回退到服务器操作系统计算机上的用户配置文件。

请查看打印机属性保留策略设置，确定是否存在以下情况：

- 是否使用了不允许用户在用户设备上存储打印机属性的旧版插件。
- 是否在 Windows 网络上使用了强制配置文件并希望保留用户的打印机配置文件。

置备打印机

May 04, 2016

有三种打印机置备方法：

- [Citrix 通用打印服务器](#)
- [自动创建的客户端打印机](#)
- [将网络打印机分配给用户](#)

在确定适用于您的环境的最佳打印解决方案时，请考虑以下事项：

- 通用打印服务器提供的以下功能不适用于 Windows 打印提供程序：图像与字体缓存、高级压缩、优化和 QoS 支持。
- 通用打印驱动程序支持由 Microsoft 定义的独立于设备的公共设置。如果用户需要访问特定于打印驱动程序制造商的设备设置，最佳解决方案可能是与 Windows 本机驱动程序配对的通用打印服务器。使用此配置，您可以在保留通用打印服务器优势的同时，允许用户使用专用打印机的功能。需要考虑的一个平衡点是，Windows 本机驱动程序需要维护。
- Citrix 通用打印服务器为网络打印机提供通用打印支持。通用打印服务器使用通用打印驱动程序，该驱动程序是服务器操作系统计算机上的单个驱动程序，允许从任何设备（包括瘦客户端和平板电脑）进行本地打印或网络打印。

要将通用打印服务器与 Windows 本机驱动程序结合使用，请启用通用打印服务器。默认情况下，如果 Windows 本机驱动程序可用，请使用 Windows 本机驱动程序。否则，将使用通用打印驱动程序。要指定对该行为的更改，例如仅使用 Windows 本机驱动程序或仅使用通用打印驱动程序，请更新通用打印驱动程序使用策略设置。

安装 Citrix 通用打印服务器 (UPS)

注意：

- 不支持在 32 位操作系统上部署通用打印服务器。
- 打印操作期间进行用户身份验证需要将通用打印服务器连接到与远程桌面服务 VDA 相同的域。
- UPClient 组件属于 VDA 安装的一部分；不需要手动安装客户端组件。

要安装 Citrix 通用打印服务器，请执行以下操作：

1. 在 Windows 2008 R2 SP1、Windows Server 2012 或 Windows Server 2012 R2 打印服务器上，在 LTSR metainstaller 中，单击“Extend Deployment”（展开部署）>“Universal Print Server”（通用打印服务器）并按照屏幕上的说明进行操作。
2. 重新启动服务器。

针对客户端打印机提供以下通用打印解决方案：

- **Citrix 通用打印机** – 在会话开始时创建的通用打印机，未绑定到打印设备。在登录期间枚举可用的客户端打印机不需要 Citrix 通用打印机，这样可以大大降低资源使用情况并降低用户登录次数。通用打印机可以打印到任何客户端打印设备。Citrix 通用打印机不一定适用于您环境中的所有用户设备或 Receiver。Citrix 通用打印机需要 Windows 环境，不支持 Citrix 脱机插件或者流传输到客户端的应用程序。对于此类环境，请考虑使用自动创建的客户端打印机和通用打印驱动程序。

要对非 Windows Receiver 使用通用打印解决方案，请使用基于 PostScript/PCL 并自动安装的其他通用打印驱动程序之一。

- **Citrix 通用打印驱动程序** – 独立于设备的打印机驱动程序。如果配置 Citrix 通用打印驱动程序，则系统默认使用基于 EMF 的通用打印驱动程序。

此外，与旧版或较低级的打印机驱动程序相比，Citrix 通用打印驱动程序可能会创建更小的打印作业。但是，可能需要特定于设备的驱动程序才能优化专用打印机的打印作业。

配置通用打印 - 使用以下 Citrix 策略设置配置通用打印。有关详细信息，请参阅屏幕上的策略设置帮助。

- 通用打印。指定何时使用通用打印。
- 自动创建一般通用打印机。允许或禁止在使用与通用打印兼容的用户设备时为会话自动创建一般 Citrix 通用打印机对象。默认情况下不自动创建一般通用打印机对象。
- 通用驱动程序首选项。指定系统尝试使用通用打印驱动程序顺序，从列表中的第一项开始。可以添加、编辑或删除驱动程序以及更改列表中驱动程序的顺序。
- 通用打印预览首选项。指定是否使用自动创建的打印机或一般通用打印机的打印预览功能。
- 通用打印 EMF 处理模式。控制在 Windows 用户设备上处理 EMF 后台打印文件的方法。默认情况下，系统将 EMF 记录直接后台处理到打印机中。借助直接后台处理到打印机中的方式，后台处理程序可以更快地处理记录，且使用的 CPU 资源更少。

有关更多策略，请参阅[优化打印性能](#)。要更改默认设置（例如纸张大小、打印质量、色彩、双面打印和份数），请参阅 [CTX113148](#)。

在用户设备中自动创建打印机 - 默认情况下，在会话开始时，系统会在用户设备上自动创建所有打印机。您可以控制为用户置备的打印机类型（如果有），并阻止自动创建。

Citrix 策略设置自动创建客户端打印机用于控制自动创建。您可以指定以下内容：

- 在每个会话开始时自动创建对用户设备可见的所有打印机，包括网络打印机和本机连接的打印机（默认值）
- 自动创建以物理方式连接到用户设备的所有本地打印机
- 仅自动创建用户设备的默认打印机
- 对所有客户端打印机禁用自动创建

自动创建客户端打印机设置要求将客户端打印机重定向设置为允许（默认值）。

默认情况下，会话开始时会在用户设备上自动创建网络打印机。系统使您能够通过指定要在每个会话中创建的网络打印机，减少枚举或映射的网络打印机数量。这类打印机称为会话打印机。

您可以按要提供邻近打印的 IP 地址过滤会话打印机策略。通过邻近打印，指定 IP 地址范围内的用户可以自动访问该范围内存在的网络打印设备。邻近打印由 Citrix 通用打印服务器提供，并且不需要进行本部分所述的配置。

邻近打印可能涉及以下情况：

- 内部公司网络与为用户自动指定 IP 地址的 DHCP 服务器一起运行。
- 公司内的所有部门均具有唯一的指定 IP 地址范围。
- 网络打印机存在于每个部门的 IP 地址范围内。

如果配置了邻近打印，则员工从一个部门转移到另一个部门时，无需进行其他打印设备配置。只要用户设备在新部门的 IP 地址范围内得以识别，即对该范围内的所有网络打印机具有访问权限。

配置要在会话中重定向的特定打印机 - 要创建由管理员分配的打印机，请配置 Citrix 策略设置会话打印机。使用以下方法之一向该策略添加网络打印机：

- 使用格式 \\servername\printername 输入打印机 UNC 路径。
- 浏览到网络上的打印机位置。
- 浏览特定服务器上的打印机。使用格式 \\servername 输入服务器名称，并单击浏览。

重要：服务器将合并所有已应用策略的所有已启用会话打印机设置，合并顺序为按优先级从最高到最低。如果在多个策略对象

中配置了某个打印机，则仅采用配置了该打印机且具有最高优先级的策略对象中的自定义默认设置。
根据会话启动时所处的位置（通过对子网等对象进行过滤），使用会话打印机设置创建的网络打印机可能有所不同。

为会话指定默认网络打印机 - 默认情况下，用户的主打印机将用作会话的默认打印机。使用 Citrix 策略设置默认打印机更改会话中在用户设备上建立默认打印机的方式。

1. 在默认打印机设置页面上，为选择客户端的默认打印机选择一项设置：
 - 网络打印机名称。此菜单中会显示使用会话打印机策略设置添加的打印机。选择用作该策略的默认打印机的网络打印机。
 - 不调整用户的默认打印机。使用默认打印机当前的终端服务或 Windows 用户配置文件设置。有关详细信息，请参阅屏幕上的策略设置帮助。
2. 将该策略应用于要施加影响的用户组（或其他过滤的对象）。

配置邻近打印 - Citrix 通用打印服务器还提供了邻近打印，这不需要本部分中所述的配置。

1. 为每个子网（或针对打印机位置）创建一个单独的策略。
2. 在每个策略中，将位于该子网所处地理位置的打印机添加到会话打印机设置。
3. 将默认打印机设置设置为不调整用户的默认打印机。
4. 按照客户端 IP 地址过滤策略。请确保更新这些策略，以反映对 DHCP IP 地址范围的更改。

维护打印环境

May 04, 2016

维护打印环境包括：

- 管理打印机驱动程序
- 优化打印性能
- 显示打印机和管理打印队列

为了最大程度地降低管理开销和打印驱动程序出现问题的可能性，Citrix 建议使用 Citrix 通用打印驱动程序。

默认情况下，如果自动创建失败，系统会安装 Windows 提供的 Windows 本机打印机驱动程序。如果驱动程序不可用，系统将回退到通用打印驱动程序。有关打印机驱动程序默认值的详细信息，请参阅[最佳做法](#)、[安全注意事项](#)和[默认操作](#)。

如果 Citrix 通用打印驱动程序并不适用于所有方案，请映射打印机驱动程序以最大程度减少服务器操作系统计算机上安装的驱动程序数量。此外，通过映射打印机驱动程序，您可以执行以下操作：

- 允许指定的打印机仅使用 Citrix 通用打印驱动程序
- 允许或阻止使用指定的驱动程序创建打印机
- 使用性能良好的打印机驱动程序替换过时或已损坏的驱动程序
- 使用 Windows 服务器上可用的驱动程序替换客户端驱动程序名称

阻止自动安装打印机驱动程序 - 应禁用自动安装打印驱动程序，以确保服务器操作系统计算机之间的一致性。可以通过 Citrix 和/或 Microsoft 策略实现这一点。要阻止自动安装 Windows 本机打印机驱动程序，请禁用 Citrix 策略设置自动安装现有的打印机驱动程序。

映射客户端打印机驱动程序 - 每个客户端都会在登录期间提供有关客户端打印机的信息（包括打印机驱动程序名称）。在自动创建客户端打印机期间，会选择与客户端提供的打印机型号名称相对应的 Windows 服务器打印机驱动程序名称。然后，自动创建过程会使用已识别的可用打印机驱动程序构建重定向的客户端打印队列。

以下是定义驱动程序替换规则以及编辑映射客户端打印机驱动程序的打印设置的常规过程：

1. 要指定自动创建的客户端打印机的驱动程序替换规则，可以通过以下方法配置 Citrix 策略设置打印机驱动程序映射和兼容性：添加客户端打印机驱动程序名称，然后从查找打印机驱动程序菜单中选择要替换客户端打印机驱动程序的服务器驱动程序。可以在此设置中使用通配符。例如，要强制所有 HP 打印机使用特定的驱动程序，可以在策略设置中指定 HP*。
2. 要禁用打印机驱动程序，请选择驱动程序名称并选中不创建设置。
3. 根据需要，编辑现有映射，删除映射，或更改列表中驱动程序条目的顺序。
4. 要编辑映射客户端打印机驱动程序的打印设置，请选择打印机驱动程序，单击设置，然后指定打印质量、方向和颜色等设置。如果指定打印机驱动程序不支持的打印选项，该选项将不起任何作用。此设置将覆盖用户在先前会话期间设置的保留打印机设置。
5. Citrix 建议在映射驱动程序之后详细测试打印机的行为，因为某些打印机功能仅在特定的驱动程序中提供。

当用户登录时，系统将在设置客户端打印机前检查客户端打印机驱动程序兼容性列表。

要优化打印性能，请使用通用打印服务器和通用打印驱动程序。以下策略可控制打印优化和压缩：

- 通用打印优化默认值。指定在为会话创建通用打印机时所使用的通用打印机默认设置：
 - 所需图像质量指定应用到通用打印的默认图像压缩限制。默认情况下，标准质量处于启用状态，这意味着用户只能使用标准或降低质量的压缩级别来打印图像。

- 启用超级压缩用于启用或禁用超出由“所需图像质量”所设置的压缩级别上减少带宽，而不降低图像质量。默认情况下，超级压缩功能处于禁用状态。
- 图像与字体缓存设置指定是否缓存在打印流中多次出现的图像和字体，以确保每个唯一的图像或字体只发送给打印机一次。默认情况下将缓存嵌入式图像和字体。
- 允许非管理员修改这些设置指定用户是否可以更改会话内的默认打印优化设置。默认情况下，不允许用户更改默认打印优化设置。
- 通用打印图像压缩限制。定义通过通用打印驱动程序所打印的图像可使用的最高质量和最低压缩级别。默认情况下，图像压缩限制设置为最佳质量(无损压缩)。
- 通用打印打印质量限制。指定在会话中生成打印输出时可用的最高分辨率 (dpi)。默认情况下，指定“无限制”。

默认情况下，发往网络打印机的所有打印作业都会从服务器操作系统计算机通过网络直接路由到打印服务器。如果网络出现时间延迟或者带宽有限，请考虑通过 ICA 连接路由打印作业。要执行此操作，请禁用 Citrix 策略设置直接连接到打印服务器。通过 ICA 连接发送的数据会进行压缩，因此通过 WAN 传输数据占用的带宽更少。

通过限制打印带宽提升会话性能 - 当文件从服务器操作系统计算机打印到用户打印机时，其他虚拟通道（例如视频）可能会因为争用带宽而导致性能下降，特别是当用户通过速度较慢的网络访问服务器时。为避免出现此类性能下降，可以限制用户打印所用的带宽。通过限制打印的数据传输速率，可将 HDX 数据流中的更多带宽用于视频、击键以及鼠标数据的传输。

重要：打印机带宽限制始终会强制执行，即使其他通道处于不使用状态时也是如此。
使用以下 Citrix 策略带宽打印机设置来配置打印带宽会话限制。要为站点设置限制，请使用 Studio 执行此任务。要为单个服务器设置限制，请在每台服务器操作系统计算机上使用 Windows 中的组策略管理控制台从本地执行此任务。

- 打印机重定向带宽限制设置指定用于打印的带宽，以千字节/秒 (kbps) 为单位。
 - 打印机重定向带宽限制百分比设置可将用于打印的带宽限制为可用总带宽的一定百分比。
- 注意：要使用打印机重定向带宽限制百分比设置以百分比形式指定带宽，还需启用总会话带宽限制。
如果为这两个设置都输入了值，将采用最严格的设置（即值较低的设置）。

要获取有关打印带宽的实时信息，请使用 Citrix Director。

下表总结了在您的环境中可以显示打印机以及管理打印队列的位置。

	打印途径	是否启用了 UAC ?	位置
客户端打印机（连接到用户设备的打印机）	客户端打印途径	On（启用）	Microsoft 管理控制台中的打印管理管理单元
		Off（禁用）	Windows 8 之前的系统：控制面板 Windows 8：打印管理管理单元
网络打印机（网络打印服务器上的打印机）	网络打印途径	On（启用）	打印服务器 > Microsoft 管理控制台中的打印管理管理单元
		Off（禁用）	打印服务器 > 控制面板
网络打印机（网络打印服务器上的打印机）	客户端打印途径	On（启用）	打印服务器 > Microsoft 管理控制台中的打印管理管理单元

	打印途径	Off (禁用) 是否启用了 UAC?	Windows 8 之前的系统：控制面板 位置 Windows 8：打印管理管理单元
本地网络服务器打印机（来自网络打印服务器且已添加到服务器操作系统计算机）	网络打印 途径	On (启用)	打印服务器 > 控制面板
		Off (禁用)	打印服务器 > 控制面板

注意：使用网络打印途径的网络打印机的打印队列是专用的，不能通过系统进行管理。

许可

Nov 02, 2016

注意

Studio and Director do not support Citrix License Server VPX. For more information about Citrix License Server VPX, see the Citrix Licensing documentation.

From Studio, you can manage and track licensing, if the license server is in the same domain as Studio or in a trusted domain. For information about other licensing tasks, see

— *Licensing Your Product*

You must be a full license administrator to complete the tasks described below, except for viewing license information. To view license information in Studio, an administrator must have at least the Read Licensing Delegated Administration permission; the built-in Full Administrator and Read-Only Administrator roles have that permission.

The following table lists the supported editions and license models:

Products	Editions	License models
XenApp	- Platinum - Enterprise - Advanced	Concurrent
XenDesktop	- Platinum - Enterprise - App - VDI	- User/Device - Concurrent

To view license information, in the Studio navigation pane, select Configuration and then Licensing. A summary of license usage and settings for the site is displayed with a list of all the licenses currently installed on the specified license server.

To manage licensing, in the Studio navigation pane, select Configuration and then Licensing. Then:

- To download a license from Citrix:
 - In the Actions pane, select Allocate Licenses.
 - Type the License Access Code, which is supplied in an email from Citrix.
 - Select a product and click Allocate Licenses. All the licenses available for that product are allocated and downloaded.

After you allocate and download all the licenses for a specific License Access Code, you cannot use that License Access Code again. To perform additional transactions with that code, log on to My Account.
- To add licenses that are stored on your local computer or on the network:
 - In the Actions pane, select Add Licenses.
 - Browse to a license file and add it to the license server.
- To change the license server:
 - In the Actions pane, select Change License Server.

2. Type the address of the license server in the form name:port, where name is a DNS, NetBIOS, or IP address. If you do not specify a port number, the default port (27000) is used.
- To select the type of license to use:
 - When configuring the Site, after you specify the license server, you are prompted to select the type of license to use. If there are no licenses on the server, the option to use the product for a 30-day trial period without a license is automatically selected.
 - If there are licenses on the server, their details are displayed and you can select one of them. Or, you can add a license file to the server and then select that one.
 - To change the product edition and licensing model:
 1. In the Actions pane, select Edit Product Edition.
 2. Update the appropriate options.
 - To access the License Administration Console, in the Actions pane, select License Administration Console. The console either appears immediately, or if the dashboard is configured as password-protected, you are prompted for License Administration Console credentials. For details about how to use the console, see
— Licensing Your Product
 .
 - To add a licensing administrator:
 1. In the middle pane, choose the Licensing Administrators tab.
 2. In the Actions pane, select Add licensing administrator.
 3. Browse to the user you want to add as an administrator and choose permissions.
 - To edit or delete a licensing administrator, When you select an administrator, the options to Edit licensing administrator (to change the administrator permissions for that administrator) and Delete licensing administrator appear in the Actions pane.
 1. In the middle pane, choose the Licensing Administrators tab and select the administrator you want to delete or edit.
 2. In the Actions pane, select either Edit licensing administrator or Delete licensing administrator.
 - To add a licensing administrator group:
 1. In the middle pane, choose the Licensing Administrators tab.
 2. In the Actions pane, select Add licensing administrator group.
 3. Browse to the group you want to act as licensing administrators and choose permissions. Adding an Active Directory Group gives licensing administrator permissions to the users within that group.
 - To edit or delete a licensing administrator group:
 1. In the middle pane, choose the Licensing Administrators tab and select the administrator group you want to delete or edit. When you select a licensing administrator group, the options to Edit licensing administrator group (to change the administrator permissions for that group) and Delete licensing administrator group appear in the Actions pane..
 2. In the Actions pane, select either Edit licensing administrator group or Delete licensing administrator group.

连接和资源

May 04, 2016

在创建站点时，将创建与托管资源的第一个连接。之后，可以更改此连接并创建新连接。只读权限管理员可以查看连接和资源详细信息；只有完全权限管理员才可以执行连接和资源管理任务。

创建连接和资源

在创建连接之前，托管资源必须可用。

1. 在 Studio 导航窗格中，选择配置 > 托管。
2. 在“操作”窗格中，选择添加连接和资源。
3. 选择新建连接。
4. 在连接页面上：
 - 选择连接类型并输入连接名称 - 选择一个有助于管理员确定主机类型和部署地址的名称。其他所需信息取决于选择的连接类型。

连接类型	所需信息
Citrix XenServer、Microsoft System Center Virtual Machine Manager、VMware vSphere 或 Microsoft Configuration Manager 局域网唤醒	输入连接 URL、用户名和密码。 ● 对于 XenServer，Citrix 建议使用 HTTPS 保证通信安全。要使用 HTTPS，必须将与 XenServer 一起安装的默认 SSL 证书替换为由可信证书颁发机构颁发的证书；请参阅 CTX128656。 ● 对于 XenServer，可以编辑新连接并选择要使用的高可用性虚拟机管理程序（如果 XenServer 上启用了高可用性）。
Citrix CloudPlatform 或 Amazon Web Services (AWS)	输入连接 URL、API 密钥和密钥。 ● 可以浏览到云管理员提供的导入密钥文件来填写 API 密钥和密钥。 ● root AWS 帐户的凭据文件（从 AWS 控制台检索）的格式与为标准 AWS 用户下载的凭据文件的格式不同。因此，Studio 不能使用此文件来填充 API 密钥和密钥字段。在 AWS 环境中使用 Studio 时，请确保使用的是 AWS IAM 凭据文件。

- 选择创建虚拟机所要使用的工具。对于提供 GPU 资源的虚拟机管理程序，请选择 Studio 工具。
5. 在存储页面上，选择存储类型和设备。使用 Machine Creation Services 时，请为新虚拟机选择网络和存储资源。如果使用 XenServer 连接上的共享存储，则可以启用 IntelliCache 来降低存储设备上的负载。有关使用 IntelliCache 的信息，请参阅下文。
 6. 如果连接具有 GPU 功能，请选择使用图形虚拟化的选项，然后选择 GPU 类型和组。
 7. 输入资源的名称。

通过现有连接创建连接和资源

1. 在 Studio 导航窗格中，选择配置 > 托管。
2. 在“操作”窗格中，选择添加连接和资源。
3. 选择使用现有连接，然后选择相关连接。
4. 选择创建虚拟机所要使用的工具。对于提供 GPU 资源的虚拟机管理程序，请选择 Studio 工具。如果连接具有 GPU 功能，请选择使用图形虚拟化的选项，然后选择 GPU 类型和组。
5. 输入资源的名称。

添加存储

1. 在 Studio 导航窗格中，选择配置 > 托管。
2. 选择连接，然后在“操作”窗格中选择添加存储。
3. 选择要添加的存储。

编辑存储

1. 在 Studio 导航窗格中，选择配置 > 托管。
2. 在某个连接下选择一个资源条目，然后在“操作”窗格中选择“编辑存储”。
3. 在标准存储页面上，选中或取消选中用于存储虚拟机的存储位置对应的复选框。如果取消选择用于接受新虚拟机的存储位置，该位置将不再接受新虚拟机。现有虚拟机将继续使用该位置（并向其中写入数据）；因此，即使在某个存储位置停止接受新虚拟机后，该位置也有可能满载。
如果使用 PVD 存储，请同时选中或取消选中 PVD 存储页面上的相应复选框。

编辑连接

请勿使用此过程来重命名连接或创建新连接。它们属于不同的操作。

1. 在 Studio 导航窗格中，选择配置 > 托管。
2. 选择连接，然后在“操作”窗格中选择编辑连接。
 - 要更改连接地址和凭据，请在连接属性页面上，单击编辑设置，然后输入新信息。
无法更改连接的 GPU 设置，因为访问此资源的计算机目录必须使用恰当的 GPU 特定的主映像。创建新连接。
 - 要为 XenServer 连接指定高可用性服务器，请在连接属性页面上，单击编辑高可用性服务器。Citrix 建议选择池中的所有服务器，以便在池主服务器出现故障时允许与 XenServer 实现通信。
 - 对于 Microsoft System Center Configuration Manager (ConfMgr) 局域网唤醒连接，请在高级页面上，输入 ConfMgr 唤醒代理、幻数据包和数据包传输信息。
 - 要基于针对连接执行的同步操作的阈值对限制进行配置，在电源管理设置允许同时启动的计算机过多或过少时，这很有帮助。
 - 在高级页面上，对于同步操作(所有类型)和 Simultaneous Personal Storage inventory updates（同步个人存储清单更新），指定两个值：一个是可在此连接上同时发生的最大绝对值，另一个是占使用此连接的所有计算机的百分比。绝对值和百分比值必须都指定，但应用的实际限制是所配置值中的较小者。
例如，在包含 34 台计算机的部署中，如果同步操作(所有类型)设置为绝对值 10 且百分比值为 10，则所应用的实际限制为 3（即，34 的 10% 四舍五入到最接近的整数，此值小于 10 台计算机这一绝对值）。
 - 指定每分钟执行的最大新操作数量。这是一个绝对数。
注意：在高级页面上的连接选项字段中输入信息时，请务必在 Citrix 技术支持代表的指导下进行。

打开或关闭连接的维护模式

打开连接的维护模式可防止任何新电源操作影响此连接上存储的任何计算机。计算机处于维护模式时，用户无法连接到计算机。如果已经连接用户，维护模式将在其注销后生效。

1. 在 Studio 导航窗格中，选择配置 > 托管。
2. 选择连接。要打开维护模式，请在“操作”窗格中选择打开维护模式。要关闭维护模式，请选择关闭维护模式。

另外，也可以针对单个计算机打开或关闭维护模式，请参阅下文。

删除连接

警告：如果删除连接，会导致大量计算机被删除，并会导致数据丢失。请确保受影响计算机上的用户数据已经备份，或者已不再需要。

删除连接之前，应确保：

- 所有用户都已从该连接上所存储的计算机中注销。
- 没有仍在运行的已断开连接的用户会话。
- 已为池计算机和专用计算机打开维护模式。
- 计算机目录中的所有计算机均已关闭。

1. 在 Studio 导航窗格中，选择配置 > 托管。
2. 选择连接，然后在“操作”窗格中选择删除连接。
3. 如果此连接上存储了计算机，系统会询问您是否删除这些计算机。如果要将其删除，请指定应对关联的 Active Directory 计算机帐户执行的操作。
删除计算机目录引用的连接时，该目录会变为不可用。如果有目录引用此连接，可以选择删除该目录。删除目录前，请确保其他连接未使用此目录。

重命名连接

1. 在 Studio 导航窗格中，选择配置 > 托管。
2. 选择连接，然后在“操作”窗格中选择重命名连接。

查看连接上的计算机详细信息

1. 在 Studio 导航窗格中，选择配置 > 托管。
2. 选择连接，然后在“操作”窗格中选择查看计算机。

上方窗格列出通过此连接访问的计算机。选择某个计算机可在下方窗格查看其详细信息。对于打开的会话，还会提供会话详细信息。

使用搜索功能可快速查找计算机。从窗口顶部的列表中选择保存的搜索，或者创建新的搜索。可以通过键入完整或部分计算机名称进行搜索，也可以构建表达式进行高级搜索。要构建表达式，请单击展开，然后从属性和运算符列表中进行选择。

管理连接上的计算机

1. 在 Studio 导航窗格中，选择配置 > 托管。
2. 选择连接，然后在“操作”窗格中选择查看计算机。
3. 在“操作”窗格中，选择以下选项之一。某些操作可能不可用，具体取决于计算机状态和连接主机类型。
 - 启动 - 启动计算机（如果计算机已关闭或挂起）。
 - 挂起 - 不关闭但暂停计算机并刷新计算机列表。
 - 关闭 - 请求关闭操作系统。
 - 强制关闭 - 强行关闭计算机并刷新计算机列表。
 - 重新启动 - 请求关闭操作系统，然后再次启动计算机。如果操作系统无法关闭，则计算机仍保持当前状态。
 - Enable maintenance mode（启用维护模式） - 要暂时停止与某个计算机的连接，请将计算机置于维护模式。在此状态下用户无法连接计算机。如果已经连接用户，则维护模式会在其注销后生效。
要为通过某个连接进行访问的所有计算机打开或关闭维护模式，请参阅上文。
 - Remove from Delivery Group（从交付组中删除） - 将计算机从交付组中删除并不会将其从交付组所用的计算机目录中删除。仅当计算机未连接任何用户时，才能将其删除（在删除计算机时，可打开维护模式暂时阻止用户连接此计算机）。
 - 删除 - 删除计算机后，用户将不再具有访问该计算机的权限，该计算机将从计算机目录中删除。删除计算机之前，应确保所有用户数据都已备份，或者不再需要这些数据。仅当计算机未连接任何用户时，才能将其删除（在删除计算机时，可打开维护模式暂时停止用户连接此计算机）。

对于涉及关闭计算机的操作，如果计算机在 10 分钟内未关闭，则会关机。如果 Windows 尝试在关闭期间安装更新，可能面临在更新完成前计算机关机的风险。

删除、重命名或测试资源

1. 在 Studio 导航窗格中，选择配置 > 托管。
2. 选择资源，然后在“操作”窗格中选择相应的条目：删除资源、重命名资源或测试资源。

将 IntelliCache 用于 XenServer 连接

通过使用 IntelliCache，托管的 VDI 部署将更节省成本，因为您可以将共享存储与本地存储结合使用。这会提高性能并降低网络流量。本地存储对共享存储的主映像进行缓存，从而减少了共享存储上的读操作数量。对于共享桌面，对不同磁盘写入的内容将写入到主机上的本地存储而不是共享存储中。

- 使用 IntelliCache 时，共享存储必须为 NFS。
- Citrix 建议您使用高性能本地存储设备来保证实现最快速的数据传输。

要使用 IntelliCache，必须在此产品和 XenServer 中均启用 IntelliCache。

- 安装 XenServer 时，选择启用精简置备(XenDesktop 的优化存储)。Citrix 不支持由启用了 Intellicache 的服务器和未启用 IntelliCache 的服务器构成的混合池。有关详细信息，请参阅 XenServer 文档。
- 在 XenApp 和 XenDesktop 中，IntelliCache 默认情况下处于禁用状态。只能在创建 XenServer 连接时更改此设置；之后将无法禁用 IntelliCache。从 Studio 添加 XenServer 连接时：
 - 选择共享作为存储类型。
 - 选中 Use IntelliCache（使用 IntelliCache）复选框。

连接计时器

可以使用策略设置来配置三种连接计时器：

- 最长连接时间计时器。此设置确定保持用户设备和虚拟桌面之间连接不中断的最长持续时间。使用会话连接计时器和会话连接计时器间隔策略设置。
- 连接空闲计时器。此设置确定如果用户不输入任何内容，用户设备与虚拟桌面之间的连接可以保持多久不中断。使用会话空闲计时器和会话空闲计时器间隔策略设置。
- 断开连接计时器。此设置确定在会话注销之前，已断开连接且锁定的虚拟桌面可以保持锁定状态的时间。使用断开连接的会话计时器和断开连接的会话计时器间隔策略设置。

如果更新其中任何一项设置，请确保部署中的设置一致。

连接租用

May 04, 2016

为确保站点数据库始终可用，Citrix 建议按照 Microsoft 的高可用性最佳实践开始部署容错 SQL Server。但是，网络问题和中断可能会阻止 Delivery Controller 访问数据库，从而使用户无法连接到其应用程序或桌面。

通过连接租用功能，用户可以连接以及重新连接到其最近使用的应用程序和桌面，即使在站点数据库不可用时也能连接，补充了 SQL Server 高可用性最佳实践。

即使用户可以获得大量的已发布资源，但他们通常只使用其中的一小部分。启用连接租用后，在正常操作期间，每个 Controller 都会缓存用户最近使用的应用程序和桌面的连接（前提是数据库可用）。

在每个 Controller 上生成的租用将上载到站点数据库，以便定期同步到站点上的其他 Controller。除了租用外，每个 Controller 的缓存还会保存应用程序、桌面、图标和工作进程信息。租用及相关信息存储在每个 Controller 的本地磁盘上。如果数据库变为不可用，Controller 将进入租用连接模式，并在用户尝试从 StoreFront 连接或重新连接到最近使用的应用程序或桌面时“重播”缓存的操作。

连接会缓存为期两周的租用期。因此，如果数据库变为不可用，仍可通过 StoreFront 对用户在前两周启动的桌面和应用程序进行访问。但是，如果桌面和应用程序在上一个为期两周的租用期内未启动，则当数据库不可用时，无法对其进行访问。例如，如果某个应用程序上次是在三周前启动的，而其租用期已过，若数据库现在变为不可用，则无法启动该应用程序。对于长时间处于活动状态或者已断开连接的应用程序或桌面会话，请延长其租用期限，以便不将其视为过期。

默认情况下，连接租用会影响整个站点；但是，可以撤消特定用户的所有租用，从而阻止其在 Controller 处于租用连接模式时访问任何应用程序或桌面。可以在 Controller 的基础上应用其他几项注册表设置。

注意事项和限制

虽然连接租用可以提高连接恢复能力和用户工作效率，但需要注意与其他功能的可用性、操作和性能相关的事项。

服务器托管的应用程序和桌面以及静态（已分配）桌面支持连接租用；数据库不可用时，池 VDI 桌面或未分配桌面的用户不支持连接租用。

Controller 处于租用连接模式时：

- 管理员不能使用 Studio、Director 或 PowerShell 控制台。
- 工作区控制不可用。用户登录 Receiver 时，会话不会自动重新连接，用户必须重新启动应用程序。
- 如果在数据库不可用后立即创建新租用，但租用信息尚未在所有 Controller 中进行同步，则在数据库变为不可用后用户可能无法启动该资源。
- 服务器托管的应用程序和桌面用户使用的会话数可能会超过所配置的会话限制。例如：
 - Controller 未处于租用连接模式时，用户从一台设备（通过 NetScaler Gateway 在外部连接）启动会话，而在 Controller 处于租用连接模式时，从 LAN 上的另一台设备进行连接，这种情况下，该会话可能不会漫游。
 - 如果应用程序恰好在数据库变为不可用之前启动，重新连接会话可能会失败；在这种情况下，将启动新的会话和应用程序实例。
- 静态（已分配）桌面的电源无法管理。如果 Controller 进入租用连接模式后 VDA 关闭电源，则在恢复数据库连接之前，VDA 一直不可用，除非管理员手动打开 VDA 电源。
- 如果启用了会话预启动和会话延迟功能，则新的预启动会话将不会启动。根据配置的阈值，当数据库不可用时，预启动的会话和延迟会话将不会结束。
- 站点内的负载管理可能会受到影响。基于服务器的连接将路由到最近使用的 VDA。负载评估器（尤其是会话计数规则）可能会超额。
- 如果使用 SQL Server Management Studio 使数据库进入脱机状态，Controller 将不会进入租用连接模式。相反，应使用以

下一种 Transact-SQL 语句：

- ALTER DATABASE <数据库名称> SET OFFLINE WITH ROLLBACK IMMEDIATE
- ALTER DATABASE <数据库名称> SET OFFLINE WITH ROLLBACK AFTER <秒数>

两个语句均会取消任何挂起的事务，并导致 Controller 丢失与数据库的连接。然后，Controller 进入租用连接模式。

启用连接租用时，在两个短暂时间间隔内用户无法进行连接或重新连接：(1) 从数据库变为不可用到 Controller 进入租用连接模式；(2) 从 Controller 退出租用连接模式到数据库访问完全恢复并且已重新注册 VDA。

有关更多注意事项，请参阅 [XenDesktop 7.6 连接租用设计注意事项](#)。

配置和部署

将部署配置为接受连接租用时：

- VDA 的版本至少为 7.6，使用这些计算机的计算机目录以及交付组必须满足这一最低要求（或使用受支持的更高版本）。
- 站点数据库大小的要求将会增加。
- 每个 Controller 都需要使用额外的磁盘空间来保存缓存的租用文件。

连接租用默认为启用状态。

可以通过 PowerShell SDK 或 Windows 注册表启用或禁用连接租用。通过 PowerShell SDK，还可以删除当前租用。以下 PowerShell cmdlet 影响连接租用；有关详细信息，请参阅 cmdlet 帮助。

- Set-BrokerSite -ConnectionLeasingEnabled \$true | \$false - 启用或禁用连接租用。默认值 = \$true
- Get-BrokerServiceAddedCapability - 输出本地 Controller 的“ConnectionLeasing”。
- Get-BrokerLease - 检索所有当前租用或过滤的租用集合。
- Remove-BrokerLease - 将一个租用或过滤的租用集合标记为删除。
- Update-BrokerLocalLeaseCache - 更新本地 Controller 上的连接租用缓存。这些数据将在下次同步时重新同步。

虚拟 IP 和虚拟环回

May 04, 2016

注意：这些功能仅对 Windows Server 2008 R2 和 Windows Server 2012 R2 计算机有效。不适用于 Windows 桌面操作系统计算机。

Microsoft 虚拟 IP 地址功能为每个会话的已发布应用程序提供动态分配的唯一 IP 地址。借助 Citrix 虚拟环回功能，可以将依赖于与 localhost（默认为 127.0.0.1）通信的应用程序配置为使用 localhost 范围 (127.*) 之内的唯一虚拟环回地址。

一些应用程序（例如 CRM 和计算机电话集成 (CTI)）将 IP 地址用于寻址、许可、身份验证或其他目的，因此，这些应用程序在会话中需要使用唯一的 IP 地址或环回地址。其他应用程序可能绑定到某个静态端口，因此，在多用户环境中尝试启动应用程序的其他实例将失败，因为该端口已在使用中。要使这些应用程序能够在 XenApp 环境中正常运行，需要为每个设备设置唯一的 IP 地址。

虚拟 IP 和虚拟环回是两个独立的功能。可以使用其中一项功能，也可以同时使用两项功能。

管理员操作摘要：

- 要使用 Microsoft 虚拟 IP，请在 Windows 服务器上启用并配置此功能。
- 要使用 Citrix 虚拟环回，请在 Citrix 策略中配置两项设置。

虚拟 IP

在 Windows 服务器上启用并配置虚拟 IP 后，会话中运行的每个已配置应用程序显示为具有唯一的地址。用户可以在 XenApp 服务器上访问这些应用程序，访问方式与访问任何其他已发布应用程序的方式相同。进程在以下情况下需要虚拟 IP：

- 进程使用硬编码的 TCP 端口号
- 进程使用 Windows 套接字并需要唯一 IP 地址或指定的 TCP 端口号

确定应用程序是否需要使用虚拟 IP 地址：

1. 获得 Microsoft 提供的 TCPView 工具。此工具可以列出所有绑定特定 IP 地址和端口的应用程序。
2. 禁用“解析 IP 地址”功能，这样您看到的将是地址而不是主机名。
3. 启动应用程序，然后使用 TCPView 查看该应用程序打开了哪些 IP 地址和端口以及哪些进程名称正在打开这些端口。
4. 配置任何打开服务器的 IP 地址（0.0.0.0 或 127.0.0.1）的进程。
5. 为确保应用程序不会在其他端口上打开相同的 IP 地址，请启动该应用程序的另一实例。

Microsoft 远程桌面 (RD) IP 虚拟化的工作方式

- 必须在 Microsoft 服务器上启用虚拟 IP 地址。
例如，在 Windows Server 2008 R2 环境中，从服务器管理器，展开“远程桌面服务”>“RD 会话主机连接”以启用 RD IP 虚拟化功能，并配置设置以使用动态主机配置协议 (DHCP) 服务器基于每个会话或每个程序动态分配 IP 地址。请参阅 Microsoft 文档以了解相关说明。
- 启用此功能后，服务器将在会话启动时从 DHCP 服务器请求动态分配的 IP 地址。
- RD IP 虚拟化功能在每会话或每程序基础上将 IP 地址分配给远程桌面连接。如果为多个程序分配 IP 地址，则它们将共享每会话 IP 地址。
- 将地址分配给会话后，该会话将在进行以下调用时使用分配的虚拟地址，而不是系统的主 IP 地址：Bind、closesocket、connect、WSAConnect、WSAAccept、getpeername、getsockname、sendto、WSASendTo、WSASocketW、gethostbyaddr、getnameinfo 和 getaddrinfo

在“远程桌面”会话托管配置中使用 Microsoft IP 虚拟化功能时，在应用程序和 Winsock 函数调用之间插入“过滤器”组件，可将该应用程序绑定到特定的 IP 地址。然后，应用程序只能看到自己应该使用的 IP 地址。该应用程序对侦听 TCP 或 UDP 通信的任何尝试都会自动绑定到其分配的虚拟 IP 地址（或环回地址），并且该应用程序打开的任何原始连接都源自绑定到该应用程序

的 IP 地址。

在返回地址的函数（如 GetAddrInfo()，由 Windows 策略控制）中，如果请求本地主机 IP 地址，则虚拟 IP 将查看返回的 IP 地址并将其更改为会话的虚拟 IP 地址。尝试通过此类名称函数获得本地服务器 IP 地址的应用程序，仅会看到分配给该会话的唯一虚拟 IP 地址。此 IP 地址通常用于后续套接字调用，如 bind 或 connect。

通常，应用程序会请求绑定到一个端口以侦听地址 0.0.0.0。如果应用程序发出此请求并使用静态端口，您无法启动该应用程序的多个实例。虚拟 IP 地址功能也会在这些类型的调用中查找 0.0.0.0，然后将调用更改为侦听特定虚拟 IP 地址。这样一来，多个应用程序便可侦听同一台计算机上的同一端口，因为这些应用程序侦听的地址是各不相同的。仅当调用在 ICA 会话中进行并且虚拟 IP 地址功能处于启用状态时，才可以更改调用。例如，如果在不同会话中运行的应用程序的两个实例同时尝试绑定到所有接口 (0.0.0.0) 和特定端口（例如 9000），它们将分别绑定到 VIPAddress1:9000 和 VIPAddress2:9000，因而不会发生冲突。

虚拟环回

启用 Citrix 虚拟 IP 环回策略设置后，每个会话都可以拥有自己的通信用虚拟环回地址。如果应用程序在 Winsock 调用中使用了 localhost 地址（默认为 127.0.0.1），虚拟环回功能只将 127.0.0.1 替换为 127.X.X.X，其中 X.X.X 表示会话 ID + 1。例如，会话 ID 为 7 的地址是 127.0.0.8。万一会话 ID 超过第四个八位字节（大于 255），地址将滚动到下一个八位字节 (127.0.1.0)，直至达到最大值 127.255.255.255。

进程在以下情况下需要虚拟环回：

- 进程使用 Windows 套接字环回 (localhost) 地址 (127.0.0.1)
- 进程使用硬编码的 TCP 端口号

将**虚拟环回策略设置**应用于使用环回地址进行进程间通信的应用程序。无需执行其他配置。虚拟环回独立于虚拟 IP，因此无需配置 Microsoft 服务器。

- 虚拟 IP 环回支持。启用后，此策略设置允许每个会话有其自己的虚拟环回地址。默认情况下，此设置处于禁用状态。此功能仅适用于虚拟 IP 虚拟环回程序列表策略设置指定的应用程序。
- 虚拟 IP 虚拟环回程序列表。此策略设置指定使用虚拟 IP 环回功能的应用程序。此设置仅在启用了虚拟 IP 环回支持策略设置时有效。

相关功能

可以使用以下注册表设置来确保虚拟环回的优先级高于虚拟 IP；这称为环回优先。但是，操作时请注意以下事项：

- 环回优先仅在 Windows 2008 R2 上受支持。
- 仅在同时启用了虚拟 IP 和虚拟环回的情况下使用环回优先；否则可能会导致意外结果。
- 注册表编辑不当会导致严重问题，可能导致需要重新安装操作系统。Citrix 无法保证因“注册表编辑器”使用不当导致出现的问题能够得以解决。使用“注册表编辑器”需自担风险。在编辑注册表之前，请务必进行备份。

在应用程序所在的服务器运行注册表。

- HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\VIP（对于 32 位计算机为 HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\VIP）
- 名称：PreferLoopback，类型：REG_DWORD，数据：1
- 名称：PreferLoopbackProcesses，类型：REG_MULTI_SZ，数据：<进程列表>

辅助数据库位置

May 04, 2016

默认情况下，配置日志记录数据库和监视数据库（辅助数据库）与站点配置数据库位于同一台服务器上。最初，这三个数据库同名。Citrix 建议您在创建站点后更改辅助数据库的位置。配置日志记录数据库和监视数据库可以托管在同一服务器或不同服务器上。各个数据库的备份策略可以有所不同。

在更改配置日志记录数据库或监视数据库的位置时：

- 以前数据库中的数据不会导入到新数据库中。
- 检索日志时，不能合并来自两个数据库的日志。
- 新数据库中的第一条日志指示数据库发生更改，但不会标识以前的数据库。

在更改配置日志记录数据库或监视数据库的位置之前，请在数据库将驻留的服务器上安装支持的 Microsoft SQL Server 版本。如果需要，可设置镜像、群集或其他支持的冗余基础结构。

可以在启用强制日志记录功能时更改配置日志记录数据库的位置。

注意：您不能使用此方法更改站点配置数据库的位置。

1. 在 Studio 导航窗格中选择配置。此时会列出三个数据库的名称和地址，以及镜像服务器地址（如果已配置）。
2. 选择要指定新位置的数据库，然后在“操作”窗格中选择更改数据库。
3. 指定新 SQL Server 安装所在服务器的位置（使用下表中的格式之一）以及数据库名称。

数据库类型	输入内容	使用此数据库配置
独立或镜像	服务器名称	使用默认实例，SQL Server 使用默认端口。
	服务器名称\实例名称	使用指定的实例，SQL Server 使用默认端口。
	服务器名称.端口号	使用默认实例，SQL Server 使用自定义端口。（需要逗号。）
其他	群集名称	群集数据库。
	可用性组侦听器	始终开启的数据库。

4. 如果希望 Studio 创建数据库，请单击确定。当出现提示时，请单击确定，然后 Studio 会自动创建数据库。Studio 会尝试使用当前 Studio 用户的凭据访问数据库；如果此操作失败，系统会提示您输入数据库用户的凭据。然后，Studio 会将数据库架构上载到数据库。（凭据仅在数据库创建期间保留。）
5. 如果要手动创建数据库，请单击生成脚本。生成的脚本中包括用于手动创建数据库和镜像数据库（如果需要）的指令。在上载架构之前，请确保数据库为空，且至少有一个用户有权访问并更改该数据库。

Delivery Controller 环境

May 04, 2016

在部署中，Delivery Controller 是负责管理用户访问的服务器端组件，它还负责代理和优化连接。Controller 还提供用于创建桌面和服务器映像的 Machine Creation Services。

站点必须至少有一个 Delivery Controller。在安装首个 Controller 并创建站点后，可添加更多 Controller。在一个站点中安装多个 Controller 有两大主要优势。

- 冗余性 — 生产站点应始终确保至少拥有两个位于不同物理服务器上的 Controller，这是最佳做法。如果一个 Controller 出现故障，其他的 Controller 可以管理连接和站点。
- 可扩展性 — 随着站点活动的增长，Controller 的 CPU 使用率将提高，SQL Server 数据库活动也会增加。更多的 Controller 可以处理更多用户以及更多的应用程序和桌面请求，并且可以提升整体响应能力。

Virtual Delivery Agent (VDA) 发现 Controller 的方式

VDA 必须首先向站点的 Controller 进行注册（建立连接），然后该 VDA 才可以使用。VDA 通过检查称为 ListOfDDCs 的 Controller 列表来查找 Controller。ListOfDDCs 包括一个或多个用于将 VDA 指向站点 Controller 的 DNS 项或 IP 地址。为实现负载均衡，VDA 会自动在列表中的所有 Controller 之间分发连接。

除 ListOfDDCs 以外，ListOfSIDs 用于指明 VDA 允许哪些计算机安全 ID (SID) 作为 Controller 与其进行联系。ListOfSIDs 可用于降低 Active Directory 上的负载或避免来自受感染 DNS 服务器的潜在安全威胁。

随着在站点中添加和删除 Controller，确保所有 VDA 上的 ListOfDDCs 和 ListOfSIDs 始终包含最新的信息十分重要。如果未更新这些列表，VDA 可能会拒绝由未在列表中的 Controller 所代理的会话启动。无效的项会使虚拟桌面系统软件的启动发生延迟。要确保列表包含最新内容，您可以：

- 使用自动更新功能，以便随着 Controller 的添加或删除自动更新 ListOfDDCs 和 ListOfSIDs。默认情况下，自动更新功能处于启用状态。
- 自我管理 – 也就是说，手动更新用于标识 Controller 的策略或注册表设置。

在 ListOfDDCs 和 ListOfSIDs 中的信息可以来源于部署中的多个位置。VDA 按顺序检查以下位置，并在发现列表的第一个位置停止：

1. 为自动更新功能所维护的静态存储位置。在启用自动更新并且 VDA 在安装后首次成功完成注册后，此位置包含 Controller 信息。（此存储还包括计算机策略信息，这样可以确保在重新启动后保留策略设置。）

在安装后首次注册时，或者禁用自动更新后，VDA 将检查以下位置：

2. 策略设置 (Controller、Controller SID)。
3. 注册表中 Virtual Desktop Agent 注册表项下的 Controller 信息。VDA 安装程序首次将根据在安装 VDA 时指定的信息填充这些值。
4. 基于 OU 的 Controller 发现。这是一种为实现向后兼容性而维护的传统方法。
5. 由 Machine Creation Services 创建的 Personality.ini 文件。

如果 ListOfDDCs 指定多个 Controller，VDA 将尝试以随机顺序连接这些 Controller。ListOfDDCs 还可以包含 Controller 组，在这些组的名称中，用括号将两个或多个 Controller 项括起。VDA 将尝试连接组中的每个 Controller，然后转向 ListOfDDCs 中的其他项。

对于已从 7.0 之前版本升级的 XenDesktop 用户，自动更新功能将替代先前版本的 CNAME 功能。如有需要，可以手动重新启用 CNAME 功能；但是，要使 DNS 失真继续工作，您将无法同时使用自动更新功能和 CNAME 功能。有关重新启用 CNAME 功能的信息，请参阅 [CTX137960](#)。

选择自动更新或自我管理的注意事项

默认情况下，用于启用/禁用自动更新的策略设置处于启用状态。

以下类型的部署无法使用自动更新，而必须使用自我管理。

- 使用 Controller 组的部署。
- 出于安全原因使用 ListOfSIDs 的部署。（对于使用 ListOfSIDs 来降低 Active Directory 负载的部署，可以使用自动更新。）
- 使用 Provisioning Services 但不使用写回磁盘的部署。
- 使用 Controller 或 Controller SID 策略设置的部署。

使用自动更新

自动更新策略设置位于 Virtual Delivery Agent 类别中。

- 要启用自动更新，请启用启用控制器自动更新策略设置。默认情况下，此设置处于启用状态。
- 要禁用自动更新，请禁用启用控制器自动更新策略设置。

如果启用自动更新，在安装 VDA 时，VDA 将尝试已在安装 VDA 时指定的其中一个 Controller 值中进行注册。安装程序会将在 VDA 安装期间指定的控制器信息写入 ListOfDDC 注册表值。

VDA 注册完成后，该 VDA 所注册的控制器将向该 VDA 发送控制器完全限定的域名 (FQDN) 和安全 ID (SID) 的最新列表。VDA 会将此列表写入自动更新静态存储中。每个控制器还将每隔 90 分钟检查站点配置数据库以获取控制器信息 - 如果自上次检查后添加或删除了控制器，或如果发生策略更改，则控制器会将已更新的列表发送到其注册的 VDA 中。VDA 将接受所接收最新列表中的所有控制器的连接。

如果 VDA 接收的列表不包括其注册的控制器（换句话说，已从站点中删除该控制器），VDA 将从列表中选择控制器并重新注册。VDA 完成注册或重新注册后，会收到更新的列表。

例如：

1. 某部署中包括三个控制器：A、B 和 C。VDA 安装后，向控制器 B 注册（该控制器在 VDA 安装期间指定）。
2. 将 D 和 E 两个控制器添加到站点中。VDA 会在 90 分钟内收到更新的列表，并将接受来自控制器 A、B、C、D 和 E 的连接。（只有在重新启动 VDA 后，负载才会均衡分散到所有控制器上。）
3. 将控制器 B 从站点中删除。VDA 会在 90 分钟内收到更新的列表，因为自上次检查后控制器已发生更改。在第 1 步中安装的 VDA 已向控制器 B 注册，但该控制器已不在列表中，这样 VDA 将从当前列表（A、C、D 和 E）中选择控制器并重新注册。

自我管理

如果不使用自动更新功能，则在站点中添加、移动或删除 Delivery Controller 后，必须更新站点（或 VDA 映像）中每个 Virtual Delivery Agent (VDA) 的 Citrix 策略设置或注册表值。还可以使用组策略对象更新注册表更改。

要使用 Citrix 策略设置进行自我管理，请执行以下操作：

1. 更新控制器策略设置中指定的 FQDN 值。此策略设置位于 Virtual Delivery Agent 类别中。
2. 如果部署中还使用 ListOfSIDs，请更新在控制器 SID 策略设置中指定的 SID 值。

要使用注册表进行自我管理，请执行以下操作：

警告：注册表编辑不当会导致严重问题，可能导致需要重新安装操作系统。Citrix 无法保证因“注册表编辑器”使用不当导致出现的问题能够得以解决。使用“注册表编辑器”需自担风险。在编辑注册表之前，请务必进行备份。

1. 更新 ListOfDDCs 注册表项，该注册表项用于列出站点中所有控制器的 FQDN。（此注册表项相当于 Active Directory 站点 OU。）各个值之间以空格分隔。使用括号将控制器组括起来。

HKEY_LOCAL_MACHINE\Software\Citrix\VirtualDesktopAgent\ListOfDDCs (REG_SZ)

如果注册表位置 HKEY_LOCAL_MACHINE\Software\Citrix\VirtualDesktopAgent 包括 ListOfDDCs 和 FarmGUID 两个注册

表项，则 ListOfDDCs 用于控制器发现；如果在 VDA 安装期间指定站点 OU，则 FarmGUID 位于此位置。

2. 或者，也可以更新 ListOfSIDs 注册表项：

HKEY_LOCAL_MACHINE\Software\Citrix\VirtualDesktopAgent\ListOfSIDs (REG_SZ)

添加、删除或移动 Controller 或者移动 VDA

May 04, 2016

要添加、删除或移动 Delivery Controller，需要具有以下角色或权限：

操作	用途	服务器角色	数据库角色
创建数据库	创建合适的空数据库	dbcreator	
创建架构	创建服务特定的所有架构，并将第一个 Controller 添加到站点	securityadmin *	db_owner
添加 Controller	将 Controller（除第一个外）添加到站点	securityadmin *	db_owner
添加 Controller（镜像服务器）	将 Controller 登录信息添加到当前位于镜像数据库的镜像角色中的数据库服务器	securityadmin *	
删除 Controller	从站点中删除 Controller		db_owner
更新架构	应用架构更新或修补程序		db_owner
* 虽然在技术层面上的限制更加严格，但实际上应将 securityadmin 服务器角色视为等同于 sysadmin 服务器角色。 使用 Studio 执行这些操作时，用户帐户必须明确属于 sysadmin 服务器角色的成员。			

如果您的部署使用数据库镜像：

- 在添加、删除或移动 Controller 之前，请确保主体数据库和镜像数据库均处于运行状态。另外，如果您通过 SQL Server Management Studio 使用脚本，请在执行脚本之前启用 SQLCMD 模式。
- 要在添加、删除或移动 Controller 后验证镜像，请运行 get-configdbconnection PowerShell cmdlet，以确保在镜像的连接字符串中设置了 Failover Partner 的值。

在添加、删除或移动 Controller 后：

- 如果已启用自动更新，Virtual Delivery Agent (VDA) 将在 90 分钟内接收已更新的 Controller 列表。
- 如果未启用自动更新，请确保更新了所有 VDA 的 Controller 策略设置或 ListOfDDCs 注册表项。将 Controller 移至其他站点后，更新两个站点上的策略设置或注册表项。

添加 Controller

无法将安装了旧版软件的服务器添加到使用此版本创建的站点中。

- 在要添加的服务器上，运行安装程序并选择 Delivery Controller 以及任何其他要安装的核心组件。
- 在 Studio 中，单击加入现有部署，并输入站点地址。

删除 Controller

删除 Controller 时，不会卸载 Citrix 软件或任何其他组件；会从数据库中删除 Controller，这样它将再也无法用于代理连接和执

行其他任务。如果删除 Controller，您可以稍后将其添回到同一个站点中或添加到其他站点中。一个站点至少需要一个 Controller，因此无法删除 Studio 中列出的最后一个 Controller。

注意：确保打开 Controller 的电源，以使 Studio 能够在一小时内加载。Studio 加载要删除的 Controller 时，请在系统提示您关闭 Controller 的电源时执行此操作。

从站点中删除 Controller 时，不会删除登录数据库时使用的 Controller 登录信息。这样可以避免删除同一计算机上由其他产品的服务所使用的登录信息的可能性。如果不再需要，则必须手动删除登录信息；需要具有 securityadmin 服务器角色权限才能删除登录信息。

重要：请先将 Controller 从站点中删除，然后再将其从 Active Directory 中删除。

1. 在 Studio 导航窗格中选择配置 > Controller，然后选择要删除的 Controller。
2. 在“操作”窗格中单击删除 Controller。如果没有正确的数据库角色和权限，可以选择生成一个脚本，数据库管理员可以通过该脚本为您删除 Controller。
3. 您可能需要从数据库服务器中删除 Controller 的计算机帐户。在执行此操作之前，请检查是否有其他服务在使用该帐户。

使用 Studio 删除 Controller 之后，该 Controller 的流量可能会出现短时间的延迟，以确保当前任务正常完成。如果要在极短的时间内删除 Controller，Citrix 建议在服务器的安装位置将其关闭，或从 Active Directory 中删除该服务器。然后在该站点上重新启动其他 Controller，确保不再与删除的 Controller 进一步通信。

将 Controller 移至其他站点

无法将 Controller 移至使用此软件的早期版本创建的站点。

1. 在 Controller 当前所在的站点（旧站点）上，从 Studio 导航窗格中选择配置 > Controller，然后选择要移动的 Controller。
2. 在“操作”窗格中单击删除 Controller。如果没有正确的数据库角色和权限，可以选择生成一个脚本，数据库管理员可以通过该脚本为您删除 Controller。一个站点至少需要一个 Controller，因此无法删除 Studio 中列出的最后一个 Controller。
3. 在要移动的 Controller 上，打开 Studio，出现相应提示时重置服务，然后选择加入现有站点，并输入新站点的地址。

将 VDA 移至另一个站点

如果 VDA 是使用 Provisioning Services 置备的或者 VDA 是现有映像，您可以在升级时或者将在测试站点中创建的 VDA 映像移至生产站点时将 VDA 移至另一个站点（从站点 1 移至站点 2）。无法将使用 Machine Creation Services (MCS) 置备的 VDA 从一个站点移至另一个站点，因为 MCS 不支持更改 ListOfDDCs（一项 VDA 检查）以在 Controller 中注册；使用 MCS 置备的 VDA 始终检查与在其中创建了该 VDA 的站点关联的 ListOfDDCs。

可以通过以下两种方式将 VDA 移至另一个站点：使用安装程序或 Citrix 策略。

- 安装程序：运行安装程序并添加 Controller，在站点 2 中为控制器指定 FQDN（DNS 项）。
重要：仅当未使用 Controller 策略设置时，才在安装程序中指定 Controller。
- 组策略编辑器：以下是在站点之间移动多个 VDA 的示例。
 1. 在站点 1 中创建包含以下设置的策略，然后过滤此策略至交付组级别，以在站点间发起分阶段的 VDA 迁移。
 - 控制器 – 包含站点 2 中一个或多个控制器的 FQDN（DNS 项）。
 - 启用控制器自动更新 – 设置为禁用。
 2. 在新策略创建 90 分钟内，交付组中的每个 VDA 都将收到警告。VDA 将忽略其收到的控制器列表（因为自动更新已禁用）；它会选择在策略（其列出了站点 2 中的控制器）中指定的一个控制器。
 3. 当 VDA 在站点 2 的控制器中成功注册后，它将接收站点 2 的 ListOfDDCs 和策略信息，默认情况下，自动更新功能已启用。由于 VDA 在站点 1 中注册的控制器不在站点 2 控制器所发送的列表上，因此，VDA 将从站点 2 列表中选择控制器并重新注册。从此时开始，VDA 会从站点 2 中自动更新信息。

基于 Active Directory OU 的控制器发现

Nov 02, 2016

This Delivery Controller discovery method is supported primarily for backward compatibility, and is valid only for Virtual Delivery Agents (VDAs) for Windows Desktop OS, not VDAs for Windows Server OS. Active Directory-based discovery requires that all computers in a Site are members of a domain, with mutual trusting relationships between the domain used by the Controller and the domain(s) used by desktops. If you use this method, you must configure the GUID of the OU in each desktop registry.

To perform an OU-based Controller discovery, run the Set-ADControllerDiscovery.ps1 PowerShell script on the Controller (each Controller contains this script in the folder \$Env:ProgramFiles\Citrix\Broker\Service\Setup Scripts). To run the script, you must have CreateChild permissions on a parent OU, plus full administration rights.

When you create a Site, a corresponding Organizational Unit (OU) must be created in Active Directory if you want desktops to discover the Controllers in the Site through Active Directory. The OU can be created in any domain in the forest that contains your computers. As best practice, the OU should also contain the Controllers in the Site, but this is not enforced or required. A domain administrator with appropriate privileges can create the OU as an empty container, then delegate administrative authority over the OU to a Citrix administrator.

The script creates several essential objects. Only standard Active Directory objects are created and used. It is not necessary to extend the schema.

- A Controllers security group. The computer account of all Controllers in the Site must be a member of this security group. Desktops in a Site accept data from Controllers only if they are members of this security group. Ensure that all Controllers have the 'Access this computer from the network' privilege on all virtual desktops running the VDA. You can do this by giving the Controllers security group this privilege. If Controllers do not have this privilege, VDAs will not register.
- A Service Connection Point (SCP) object that contains information about the Site, such as the Site name. If you use the Active Directory Users and Computers administrative tool to inspect a Site OU, you might need to enable Advanced Features in the View menu to see SCP objects.
- A container called RegistrationServices, which is created in the Site OU. This contains one SCP object for each Controller in the Site. Each time the Controller starts, it validates the contents of its SCP and updates it, if necessary.

If multiple administrators are likely to add and remove Controllers after the initial installation, they need permissions to create and delete children on the RegistrationServices container, and Write properties on the Controllers security group; these permissions are granted automatically to the administrator who runs the Set-ADControllerDiscovery.ps1 script. The domain administrator or the original installing administrator can grant these permissions, and Citrix recommends setting up a security group to do this.

When you are using a Site OU:

- Information is written to Active Directory only when installing or uninstalling this software, or when a Controller starts and needs to update the information in its SCP (for example, because the Controller was renamed or because the communication port was changed). By default, the Set-ADControllerDiscovery.ps1 script sets up permissions on the objects in the Site OU appropriately, giving each Controller Write access to its SCP. The contents of the objects in the Site OU are used to establish trust between desktops and Controllers. Ensure that:
 - Only authorized administrators can add or remove computers from the Controllers security group, using the security group's access control list (ACL).

- Only authorized administrators and the respective Controller can change the information in the controller's SCP.
- If your deployment uses replication, be aware of potential delays; see the Microsoft documentation for details. This is particularly important if you create the Site OU in a domain that has domain controllers in multiple Active Directory sites. Depending on the location of desktops, Controllers, and domain controllers, changes that are made to Active Directory when you are initially creating the Site OU, installing or uninstalling Controllers, or changing Controller names or communication ports might not be visible to desktops until that information is replicated to the appropriate domain controller. The symptoms of such replication delay include desktops that cannot establish contact with Controllers and are therefore not available for user connections.
- This software uses several standard computer object attributes in Active Directory to manage desktops. Depending on your deployment, the machine object's fully qualified domain name, as stored in the desktop's Active Directory record, can be included as part of the connection settings that are returned to the user to make a connection. Ensure that this information is consistent with information in your DNS environment.

Permissions summary

To create a Site, the Citrix administrator who runs the script must have rights over the Site OU to create objects (SCP, container, and security group).

(If the Site OU is not present, the administrator must have rights to create that as well. Citrix recommends that the AD domain administrator pre-create that OU and delegate rights to it to the Citrix Site administrator identity. Optionally, the script can also create the Site OU. To allow this, the administrator needs the “create OU” right on the new OU's parent OU. However, as noted, Citrix does not recommend this.)

Later, to add or remove a Controller from the Site, the Citrix administrator must have rights to add/remove a machine from the security group, and create/delete an SCP.

During normal operations, Controllers and VDAs need read rights to all objects in the OU and below. VDAs access the OU as their own machine identity; that machine identity needs at least read rights in the OU to be able to discover Controllers. A Controller also needs the rights to set properties on its own SCP object in the container.

Granting the Citrix administrator full rights to the child OUs will permit all these actions. However, if your deployment has stricter security requirements (such as restricting who can use the script for which action), you can use the Delegation of Control wizard to set specific rights. The following example procedure grants rights to create the Site.

1. Create an OU to contain the child objects (Service Connection Point (SCP), container, and security group).
2. Select the OU, then right-click and select **Delegate Control**.
3. In the Delegation of Control wizard, specify the domain user to delegate control to for the OU.
4. On the **Tasks to Delegate** page, select **Create a custom task to delegate**.
5. On the **Active Directory Object type** page, accept the default **This folder, existing objects in this folder, and creation of new objects in this folder**.
6. On the **Permissions** page, select **Permissions Create All Child Objects**.
7. Finish the wizard to confirm the privileges.

To move a Controller to another Site using OU-based Controller discovery

Follow the directions in [Move a Controller to another Site](#). After you remove the Controller from the old Site (step 2), run the PowerShell script Set-ADControllerDiscovery –sync.

This script synchronizes the OU with the current set of Controllers. After joining the existing Site (step 3), run the same script on any Controller in the new Site.

会话管理

May 04, 2016

维护会话处于活动状态对于提供最佳用户体验至关重要。如果由于网络不稳定、网络延迟变化无常以及无线设备的覆盖范围受限等因素而使连接断开，会令用户感到沮丧。对于许多移动工作者（如医院的医护工作者）而言，首先要能够在多个工作站之间快速切换，并在每次登录时访问同一组应用程序。

使用以下功能可以优化会话的可靠性，减少不便之处、停机时间以及生产力损失，还可以为移动用户提供在设备间快速、轻松漫游的能力。

- 会话可靠性
- 客户端自动重新连接
- ICA 保持活动状态
- 工作区控制

会话可靠性

会话可靠性使会话在网络连接中断时保持活动状态并显示在用户的屏幕上。用户仍然可以看到他们正在使用的应用程序，直至网络连接恢复。

此功能对于使用无线连接的移动用户尤为有用。例如，使用无线连接的用户进入铁路隧道后将暂时失去连接。通常，会话会断开连接并从用户屏幕上消失，然后该用户必须重新连接到已断开连接的会话。会话可靠性可使会话在计算机上保持活动状态。如果连接已断开，用户的显示内容将冻结，且光标变成一个旋转的沙漏，直至用户到达通道的另一端后恢复连接。用户在连接中断期间可继续访问显示内容，在网络连接恢复后可继续与应用程序交互。会话可靠性可重新连接用户而不提示进行重新身份验证。

Citrix Receiver 用户无法覆盖 Controller 设置。

可将会话可靠性与安全套接字层 (SSL) 一起使用。SSL 仅对用户设备和 NetScaler Gateway 之间发送的数据进行加密。

使用以下策略设置启用和配置会话可靠性：

- 会话可靠性连接策略设置可允许或阻止会话可靠性。
- 会话可靠性超时策略设置的默认值为 180 秒（3 分钟）。尽管您可以延长通过会话可靠性使会话保持打开状态的时间长度（此功能的主要目标是为用户提供方便，因此，它不会提示用户重新进行身份验证）。但是，如果您延长会话保持打开状态的时间，则可能导致用户感到不耐烦而离开用户设备，从而使未经授权的用户有机会访问该会话。
- 传入会话可靠性连接使用端口 2598，除非更改会话可靠性端口号策略设置中的端口号。
- 如果您不希望用户无需重新进行身份验证即可重新连接到已中断的会话，请使用客户端自动重新连接功能。您可以配置客户端自动重新连接身份验证策略设置，以便在用户重新连接到中断的会话时提示用户重新进行身份验证。

如果您同时使用了会话可靠性和客户端自动重新连接，这两项功能将按顺序发挥作用。经过在会话可靠性超时策略设置中指定的时间长度之后，会话可靠性将关闭或断开用户会话。之后，客户端自动重新连接策略设置将生效，尝试将用户重新连接到断开连接的会话。

客户端自动重新连接

通过客户端自动重新连接功能，Receiver 可以检测到 ICA 会话的意外断开连接，并自动将用户重新连接到受影响的会话。在服务器上启用此功能后，用户无需手动进行重新连接即可继续工作。

对于应用程序会话，Receiver 将一直尝试重新连接会话，直到重新连接成功或者用户取消重新连接尝试。

对于桌面会话，Receiver 将在指定时间段内尝试重新连接会话，除非重新连接成功或者用户取消了重新连接尝试。默认情况

下，此时间段为五分钟。要更改此时间段，请在用户设备上编辑以下注册表项：

HKLM\Software\Citrix\ICA Client\TransportReconnectRetryMaxTimeSeconds; DWORD;<秒数>

其中，<秒数> 是以秒为单位的数字，经过这些秒数后，将不再尝试重新连接会话。

使用以下策略设置启用和配置客户端自动重新连接：

- 客户端自动重新连接。允许或禁止连接中断后由 Receiver 自动重新连接。
- 客户端自动重新连接身份验证。允许或禁止自动重新连接时要求用户进行身份验证。
- 客户端自动重新连接日志记录。允许或禁止在事件日志中记录重新连接事件。默认情况下，禁用日志记录。启用后，服务器的系统日志会捕获与成功和失败的自动重新连接事件有关的信息。每台服务器都会将与重新连接事件有关的信息存储在自己的系统日志中；站点不会提供所有服务器的综合性重新连接事件日志。

客户端自动重新连接包含基于加密用户凭据的身份验证机制。当用户最初登录到站点时，服务器会加密用户凭据并将其存储在内存中，然后创建并向 Receiver 发送包含加密密钥的 Cookie。Receiver 将该密钥提交给服务器以便重新连接。服务器会解密这些凭据，并将其提交到 Windows 登录以便进行身份验证。Cookie 过期后，用户必须重新进行身份验证才能重新连接到会话。

如果启用客户端自动重新连接设置，则不使用 Cookie。在 Receiver 试图自动重新连接时，用户会看到一个对话框，要求输入凭据。

要最大程度地保护用户凭据和会话，请对客户端与站点之间的所有通信使用 SSL 加密。

可以使用 `icaclient.adm` 文件对 Citrix Receiver for Windows 禁用客户端自动重新连接。有关详细信息，请参阅您的 Receiver for Windows 版本的文档。

连接设置也会影响客户端自动重新连接：

- 默认情况下，通过策略设置在站点级别启用客户端自动重新连接，如上所述。无需对用户重新进行身份验证。但是，如果将服务器的 ICA TCP 连接配置为在出现中断的通信链路时重置会话，则不会发生自动重新连接。在连接中断或超时的情况下服务器会断开与会话的连接，仅在此时客户端自动重新连接才会发挥作用。在这种情况下，ICA TCP 连接指 TCP/IP 网络上用于会话的服务器虚拟端口（而非实际的网络连接）。
- 默认情况下，服务器上的 ICA TCP 连接设置为在连接中断或超时的情况下断开会话。断开连接的会话在系统内存中保持不变，并可供 Receiver 进行重新连接。
- 可将连接配置为对中断或超时的连接重置或注销会话。如果会话重置，尝试重新连接会启动新会话；并非将用户还原到正在使用的应用程序中的同一位置，而是重新启动应用程序。
- 如果将服务器配置为重置会话，客户端自动重新连接会创建新会话。此过程要求用户输入其凭据才能登录到服务器。
- 如果 Receiver 或插件提交错误的身份验证信息，自动重新连接可能会失败。在受到攻击期间或者服务器认定距检测到断开的连接已过去了太长时间时，可能会发生这种情况。

ICA 保持活动状态

启用 ICA 保持活动状态功能可防止断开已损坏的连接。启用此功能后，此功能可防止在服务器未检测到任何活动（例如，无时钟变化、无鼠标移动、无屏幕更新）时，远程桌面服务与该会话断开连接。服务器每隔几秒钟会发送一次保持活动状态数据包，以检测会话是否处于活动状态。如果会话不再处于活动状态，服务器会将该会话标记为已断开连接。

注意：ICA 保持活动状态功能仅在不使用会话可靠性的情况下起作用。会话可靠性自身具有防止被破坏的连接被断开连接的机制。请仅为不使用会话可靠性的连接配置 ICA 保持活动状态。

ICA 保持活动状态设置将覆盖 Microsoft Windows 组策略中配置的保持活动状态设置。

使用以下策略设置启用和配置 ICA 保持活动状态：

- ICA 保持活动状态超时。指定用于发送 ICA 保持活动状态消息的间隔（1 至 3600 秒）。如果您希望网络监视软件关闭环境（其中连接很少断开，是否允许用户重新连接到会话并不重要）中处于非活动状态的连接，请勿配置此选项。

默认间隔是 60 秒：每 60 秒向用户设备发送一次 ICA 保持活动状态数据包。如果用户设备在 60 秒内没有响应，则 ICA 会话的状态将变为断开连接。

- ICA 保持活动状态。发送或阻止发送 ICA 保持活动状态消息。

工作区控制

工作区控制允许桌面和应用程序随用户从一个设备移动到另一个设备。此漫游功能使用户在登录后可从任何位置访问所有桌面或打开应用程序，而无需在每个设备中重新启动桌面或应用程序。例如，工作区控制可以帮助医院的医务人员，使他们可以在不同的工作站之间快速移动，并可在每次登录后访问同一组应用程序。如果您将工作区控制选项配置为允许上述功能，则这些工作人员可以与一个客户端设备中的多个应用程序断开连接，然后在其他客户端设备上重新连接以打开相同的应用程序。

工作区控制将影响下列活动：

- **登录** – 默认情况下，工作区控制使用户能够在登录时自动重新连接到所有正在运行的桌面和应用程序，而无需手动重新打开它们。通过工作区控制，用户可以打开已断开连接的桌面或应用程序，以及其他客户端设备上的任何活动桌面或应用程序。与桌面或应用程序断开连接后，该桌面或应用程序将继续在服务器上运行。如果您的漫游用户需要在一个客户端设备上使部分桌面或应用程序保持运行状态，同时在另一客户端设备上重新连接到部分桌面或应用程序，您可以将登录重新连接行为配置为仅打开用户先前断开连接的桌面或应用程序。
- **重新连接** – 登录到服务器后，用户可以随时单击“重新连接”来重新连接到所有桌面或应用程序。默认情况下，单击“重新连接”将打开已断开连接的桌面或应用程序，以及当前正在另一个客户端设备上运行的任何桌面或应用程序。可以将“重新连接”配置为仅打开用户先前断开连接的桌面或应用程序。
- **注销** – 对于通过 StoreFront 打开桌面或应用程序的用户，您可以将“注销”命令配置为使用户从 StoreFront 和所有活动会话一起注销，也可以将其配置为仅从 StoreFront 注销。
- **断开连接** – 用户可以一次与所有正在运行的桌面和应用程序断开连接，而无需与每个应用程序逐个断开连接。

工作区控制仅适用于通过 Citrix StoreFront 连接访问桌面和应用程序的 Receiver 用户。默认情况下，已为虚拟桌面会话禁用工作区控制，但已为托管的应用程序启用该功能。默认情况下，不会在已发布的桌面与这些桌面内部运行的任何已发布应用程序之间进行会话共享。

当用户移动到新客户端设备时，用户策略、客户端驱动器映射和打印机配置将随之进行适当更改。（用户）策略和（客户端驱动器）映射是根据用户当前登录到会话所使用的客户端设备来应用的。例如，如果医务人员从医院急救室的客户端设备注销，然后登录到该医院的 X 射线实验室的工作站，则适用于 X 射线实验室中的会话的策略、打印机映射和客户端驱动器映射将在该会话启动时生效。

您可以自定义用户位置发生变化后为其显示哪些打印机。您还可以控制用户是否可以打印到本地打印机、用户进行远程连接时消耗的带宽量，以及用户打印体验的其他方面。

有关为用户启用和配置工作区控制的信息，请参阅 StoreFront 文档。

在 Studio 中使用搜索

May 04, 2016

使用“搜索”功能可查看有关特定计算机、会话、计算机目录、应用程序或交付组的信息。

1. 在 Studio 导航窗格中选择搜索。

注意：无法使用“搜索”框在计算机目录或“交付组”选项卡中进行搜索。使用导航窗格中的“搜索”节点。
要在显示画面中显示其他搜索条件，请单击“搜索”下拉字段旁边的加号。单击减号可删除搜索条件。

2. 输入名称，或使用下拉列表选择用于查找项目的其他搜索选项。

3. （可选）选择另存为保存您的搜索。此搜索即会显示在“保存的搜索”列表中。

或者，单击“Expand Search”（展开搜索）图标（两个向下的尖括号）可显示搜索属性下拉列表；可通过从下拉列表的属性中生成表达式来执行高级搜索。

可加快搜索速度的提示：

- 要在显示画面中显示作为搜索和排序依据的其他特性，请在任意列上单击鼠标右键，然后选择选择列。
- 要查找已连接到计算机的用户设备，请使用“Client (IP)”（客户端(IP)）和“Is”（是），然后输入设备的 IP 地址。
- 要查找活动的会话，请使用 Session State（会话状态）、Is（是）和 Connected（已连接）。
- 要列出交付组中的所有计算机，请在导航窗格中选择交付组，再选择相应组，然后在“操作”窗格中选择查看计算机。

IPv4/IPv6 支持

May 04, 2016

此版本支持纯 IPv4 部署、纯 IPv6 部署，以及使用重叠 IPv4 和 IPv6 网络的双协议栈部署。

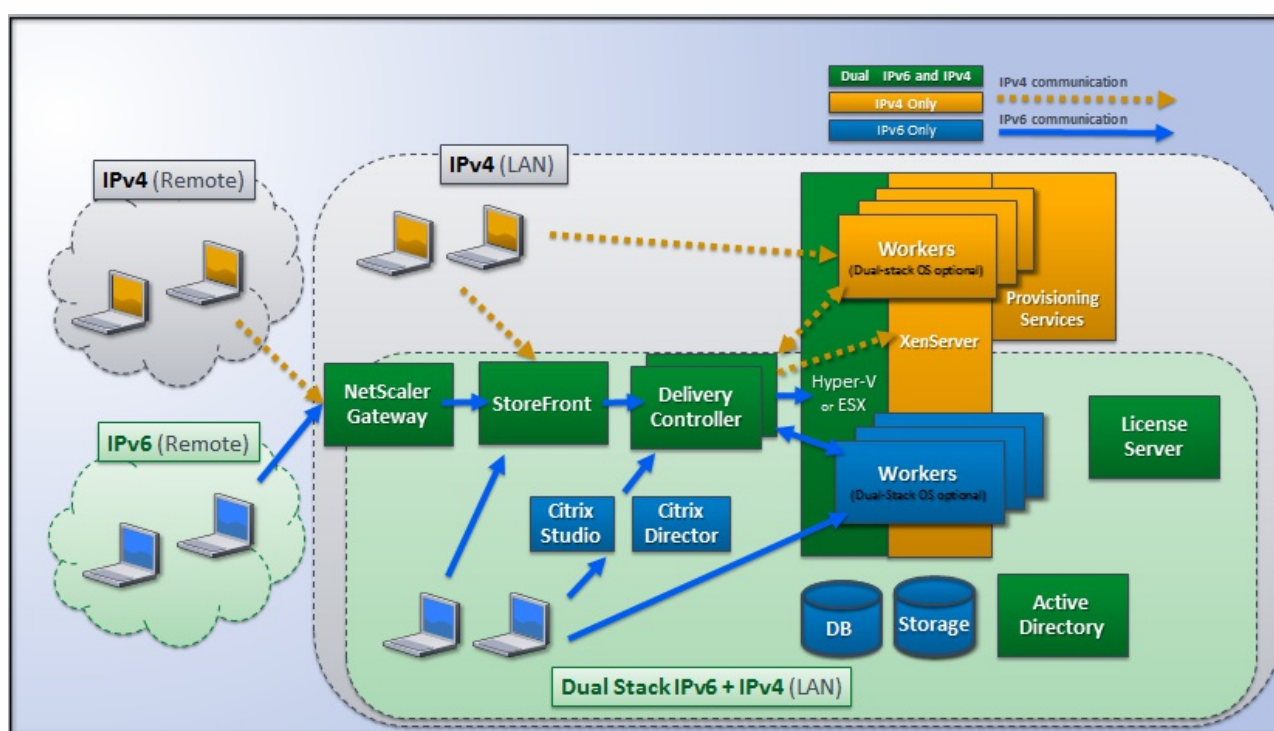
IPv6 通信通过与 Virtual Delivery Agent (VDA) 连接相关的两个 Citrix 策略设置进行控制：

- 强制使用 IPv6 的主要设置：仅使用 IPv6 控制器注册。
- 定义 IPv6 网络掩码的从属设置：控制器注册 IPv6 网络掩码。

启用仅使用 IPv6 控制器注册策略设置时，对于传入连接，VDA 将使用 IPv6 地址向 Delivery Controller 注册。

双协议栈 IPv4/IPv6 部署

下图说明了双协议栈 IPv4/IPv6 部署。在此情景中，工作者是安装在虚拟机管理程序或者物理系统上的 VDA，主要用于启用应用程序和桌面的连接。支持双 IPv6 和 IPv4 的组件在使用隧道或双协议软件的操作系统上运行。



这些 Citrix 产品、组件和功能仅支持 IPv4：

- Provisioning Services
- XenServer 6.x 版
- 不由仅使用 IPv6 控制器注册策略设置控制的 VDA
- XenApp 7.5 之前的版本、XenDesktop 7 之前的版本及 EdgeSight

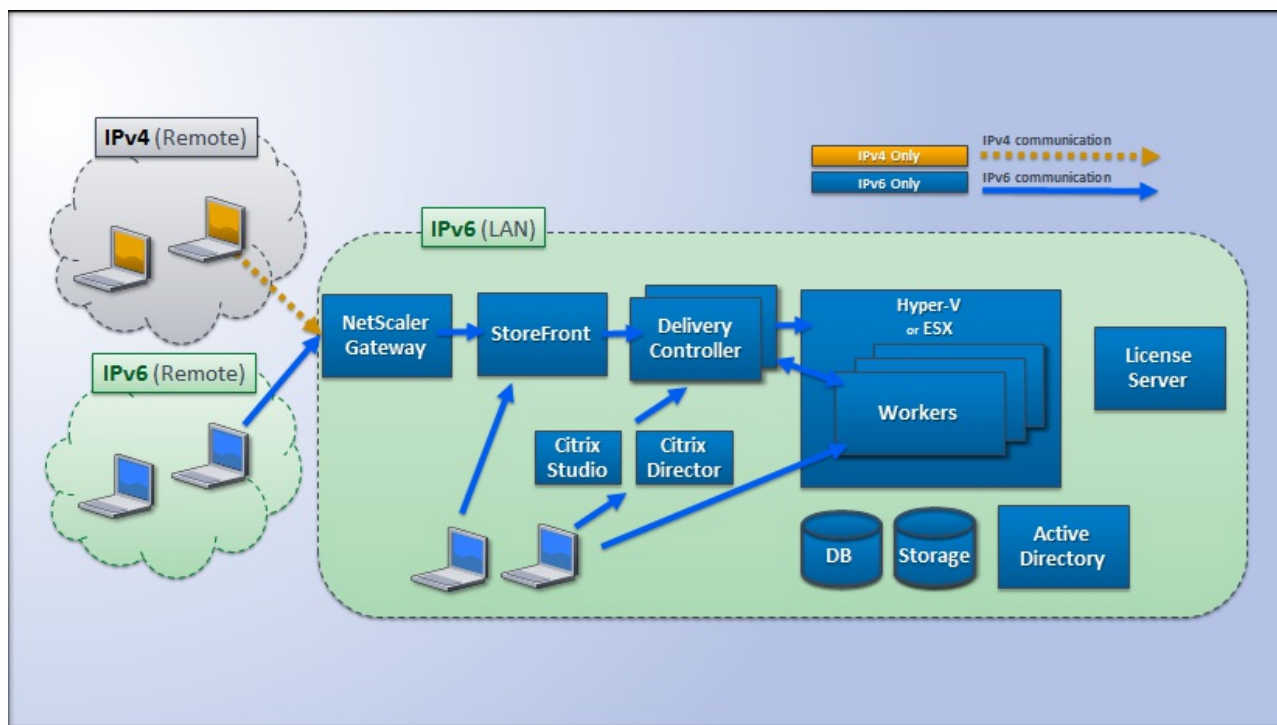
对于此部署：

- 如果一个团队经常使用 IPv6 网络，而管理员希望他们使用 IPv6 通信，管理员将基于启用了主要 IPv6 策略设置（即启用仅使用 IPv6 控制器注册）的工作者映像或组织单位 (OU) 来发布 IPv6 桌面和应用程序。
- 如果一个团队经常使用 IPv4 网络，管理员将基于关闭了主要 IPv6 策略设置（即禁用仅使用 IPv6 控制器注册）的工作者映像或 OU 来发布 IPv4 桌面和应用程序。

纯 IPv6 部署

下图说明了纯 IPv6 部署。在此情景中：

- 组件在配置为支持 IPv6 网络的操作系统上运行。
- 对所有 VDA 启用主要 Citrix 策略设置（仅使用 IPv6 控制器注册）；他们必须使用 IPv6 地址向控制器注册。



IPv6 的策略设置

有两个 Citrix 策略设置会影响对纯 IPv6 或双协议栈 IPv4/IPv6 实现的支持。请配置与连接相关的以下策略设置：

- 仅使用 IPv6 控制器注册 — 控制 Virtual Delivery Agent (VDA) 使用哪种形式的地址来向 Delivery Controller 注册。默认情况下已禁用
 - VDA 与控制器进行通信时，将按以下优先级选择使用单个 IPv6 地址：全局 IP 地址、唯一本地地址 (ULA)、链接本地地址（仅当没有其他 IPv6 地址可用时）。
 - 禁用后，VDA 将使用计算机的 IPv4 地址向控制器注册并与之通信。
- 控制器注册 IPv6 网络掩码 — 一台计算机可以具有多个 IPv6 地址；此策略设置允许管理员将 VDA 限定到首选子网而非全局 IP（如果已注册一个全局 IP）。此设置指定 VDA 将要注册的网段：VDA 仅在与指定网络掩码匹配的网段上注册。仅当启用仅使用 IPv6 控制器注册策略设置时，此设置才有效。默认值 = 空字符串

重要：VDA 使用 IPv4 还是 IPv6 完全由这些策略设置决定。换言之，要使用 IPv6 寻址，VDA 必须由启用了仅使用 IPv6 控制器注册设置的 Citrix 策略控制。

部署注意事项

如果您的环境同时包含 IPv4 和 IPv6 网络，您将需要对仅 IPv4 客户端和可以访问 IPv6 网络的客户端分开进行交付组配置。考虑使用命名、手动 Active Directory 组分配或智能访问过滤器来区分用户。

如果连接在 IPv6 网络上发起，但随后尝试从仅具有 IPv4 访问权限的内部客户端再次进行连接，可能无法重新连接到会话。

客户端文件夹重定向

May 04, 2016

客户端文件夹重定向改变了在主机端会话中访问客户端文件的方式。当仅在主机端上启用客户端驱动器映射时，客户端满载的卷会作为通用命名约定 (UNC) 链接自动映射到会话中。如果您在主机端上启用客户端文件夹重定向，同时用户也在用户设备上配置客户端文件夹重定向，将重定向用户指定的部分本地卷。

只有用户指定的文件夹会作为 UNC 链接显示在会话内，而不是显示用户设备上的完整文件系统。如果通过注册表禁用 UNC 链接，客户端文件夹将在会话内显示为映射的驱动器。

只有 Windows 桌面操作系统计算机支持客户端文件夹重定向。

分离和重新附加设备时，不保存面向外部 USB 驱动器的客户端文件夹重定向。

在主机端启用客户端文件夹定向。然后，在客户端设备上，指定要重定向的文件夹（用于指定客户端文件夹选项的应用程序包含在此版本随附的 Citrix Receiver 中）。

警告：注册表编辑不当会导致严重问题，可能导致需要重新安装操作系统。Citrix 无法保证因“注册表编辑器”使用不当导致出现的问题能够得以解决。使用“注册表编辑器”需自担风险。在编辑注册表之前，请务必进行备份。

1. 在主机端：

1. 创建注册表项 HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\Client Folder Redirection。
2. 创建 REG_DWORD 值。
 - 名称：CFROnlyModeAvailable
 - 类型：REG_DWORD
 - 数据：设置为 1

2. 在用户设备上：

1. 确保安装了最新版本的 Receiver。
2. 从 Receiver 安装目录启动 CtxCFRUI.exe。
3. 选择自定义单选按钮，并添加、编辑或删除文件夹。
4. 断开连接然后重新连接会话，以使设置生效。

Personal vDisk 7.x

May 04, 2016

Personal vDisk 功能保留了池桌面和流桌面的单映像管理功能，同时允许用户安装应用程序和更改自己的桌面设置。在涉及池桌面的传统虚拟桌面基础结构 (Virtual Desktop Infrastructure, VDI) 部署中，当管理员更改主映像时，用户将丢失自己的自定义设置和个人应用程序，而使用 Personal vDisk 功能的部署则与此不同，此部署会保留这些更改。这意味着管理员能够轻松地集中管理其主映像，同时向用户提供个性化的自定义桌面体验。

Personal vDisk 功能可以将对用户的 VM 所做的所有更改重定向到连接至用户 VM 的独立磁盘（即个人虚拟磁盘），从而将每位用户的个性化设置分隔开来。个人虚拟磁盘中的内容在运行时与主映像中的内容混合在一起，以提供一致的体验。通过这种方式，用户仍然能够访问主映像中由管理员置备的应用程序。

个人虚拟磁盘分为两个部分，这两个部分使用不同的驱动器盘符，且默认具有相同的大小：

- 用户配置文件 - 包含用户数据、文档和用户配置文件。默认情况下，这部分使用驱动器 P:，但可以在创建包含使用个人虚拟磁盘的计算机的目录时选择其他驱动器盘符。使用的驱动器还取决于 EnableUserProfileRedirection 设置。
- 虚拟硬盘 (.vhd) 文件 - 包含所有其他项目，如安装在 C:\Program Files 中的应用程序。这部分不在 Windows 资源管理器中显示，因为版本 5.6.7 不需要驱动器盘符。

Personal vDisk 支持置备部门级应用程序，以及用户下载并安装的应用程序，包括需要驱动程序（阶段 1 驱动程序除外）的应用程序、数据库和计算机管理软件。如果用户所做的更改与管理员所做的更改存在冲突，Personal vDisk 可提供一种简单的自动化方法来协调这些更改。

此外，也可以在用户的环境中置备本地管理的应用程序（例如由本地 IT 部门置备并管理的应用程序）。用户在可用性方面体会不到任何差异；Personal vDisk 功能可确保所做的所有更改以及安装的所有应用程序都存储在虚拟磁盘中。如果个人虚拟磁盘上的应用程序与主映像上的应用程序完全一致，则将丢弃个人虚拟磁盘上的备份以节省空间，但用户仍然能够访问对应的应用程序。

您可以在虚拟机管理程序上存储个人虚拟磁盘（以物理方式），但它们不必与连接到虚拟桌面的其他磁盘位于相同的位置。这样可以降低个人虚拟磁盘存储的成本。

站点创建期间，创建连接时，应为虚拟机使用的磁盘定义存储位置。可以将个人虚拟磁盘与用于操作系统的磁盘分隔开来。每台 VM 都必须对这两种磁盘的存储位置具有访问权限。如果您为这两种磁盘使用本地存储，则本地存储必须可从同一虚拟机管理程序进行访问。为确保满足此要求，Studio 仅提供兼容的存储位置。稍后，还可以从 Studio 中的“配置”>“托管”将个人虚拟磁盘以及它们的存储添加到现有主机（而不是计算机目录）。

可使用任何偏好的方法定期备份个人虚拟磁盘。虚拟磁盘是虚拟机管理程序的存储层中的标准卷，因此，您可以像备份任何其他卷一样备份它们。

Personal vDisk 7.6.1 中的新增功能

本版本中包括以下增强功能：

- 本版本的 Personal vDisk 中包含的性能增强功能缩短了对个人虚拟磁盘目录应用映像更新所需的时间。

本版本中修复了以下已知问题：

- 尝试将基础虚拟机从 Microsoft Office 2010 原位升级到 Microsoft Office 2013 导致显示一个重新配置窗口，后跟以下错误消息：Error 25004.The product key you entered cannot be used on this machine。（错误 25004。输入的产品密钥不能在此计算机上使用。）过去，建议您先卸载基础虚拟中的 Office 2010，然后再安装 Office 2013。现在，对基础虚拟机执行原位升级时不再需要卸载 Office 2010 (#391225)。

- 映像更新过程中，如果用户的个人虚拟磁盘上存在更高版本的 Microsoft .Net，该版本将被基础映像中较低版本覆盖。这会导致运行个人虚拟磁盘上安装的某些需要更高版本的应用程序（例如 Visual Studio）的用户遇到问题 (#439009)。
- 不能使用安装并启用了个人虚拟磁盘的 Provisioning Services 映像磁盘创建非个人虚拟磁盘计算机目录。此限制已删除 (#485189)。

关于 Personal vDisk 7.6

版本 7.6 中的新增功能：

- 改进了 Personal vDisk 错误处理和报告功能。在 Studio 中，显示目录中启用 PvD 的计算机时，“PvD”选项卡提供映像更新期间的监视状态，以及预计完成时间和进度。还提供了增强的状态显示。
- 早期版本的 Personal vDisk 映像更新监视工具可以从 ISO 介质 (ISO\Support\Tools\Scripts\PvdTool) 中获取。早期版本支持监视功能，但报告功能不如当前版本强大。
- Provisioning Services 测试模式允许在测试目录中使用更新的映像引导计算机。验证其稳定性后，可以将个人虚拟磁盘的测试版本提升为生产版本。
- 提供了一项新功能。利用此功能，可以计算两个清单之间的增量，而非针对每个 PvD 桌面进行计算。提供了可以导出和导入 MCS 目录的早期清单的新命令。（Provisioning Services 主虚拟磁盘已经具有之前的清单。）

版本 7.6 中修复的 7.1.3 中的已知问题：

- 中断 Personal vDisk 安装升级可能会损坏现有 Personal vDisk 安装。[#424878]
- 如果 Personal vDisk 长时间运行，并发生非分页内存泄露，虚拟桌面可能会无响应。[#473170]

版本 7.6 中的已知新问题：

- 如果存在防病毒产品，可能会影响运行清单或执行更新所需的时间。如果将 CtxPvD.exe 和 CtxPvDSvc.exe 添加到防病毒产品的 PROCESS 排除列表中，会提高性能。这些文件位于 C:\Program Files\Citrix\personal vDisk\bin 中。[#326735]
- 继承自主映像的文件之间的硬链接不会保留在 Personal vDisk 目录中。[#368678]
- 在 Personal vDisk 主映像上将 Office 2010 升级到 2013 后，Office 可能无法在虚拟机上启动，因为 Office KMS 许可产品密钥已在升级期间删除。解决方法是：卸载 Office 2010 并在主映像上重新安装 Office 2013。[#391225]
- Personal vDisk 目录不支持 VMware Paravirtual SCSI (PVSCSI) 控制器。要防止出现此问题，请使用默认控制器。[#394039]
- 对于使用 Personal vDisk 5.6.0 版本创建并升级到 7 的虚拟桌面，之前登录到主虚拟机 (VM) 的用户可能无法在其池 VM 中找到所有文件。这是因为用户登录到其池 VM 时创建了新的用户配置文件。此问题无法解决。[#392459]
- 启用 Windows 系统保护功能时，运行 Windows 7 的个人虚拟磁盘无法使用备份和还原功能。如果系统保护处于禁用状态，将会备份用户配置文件，但不对 userdata.v2.vhd 文件进行备份。Citrix 建议禁用系统保护并使用备份和还原功能来备份用户配置文件。[#360582]
- 使用磁盘管理工具在基础 VM 上创建 VHD 文件时，可能无法装载 VHD。解决方法为，将 VHD 复制到 PvD 卷。[#355576]
- 删除 Office 2010 软件后，其快捷方式仍保留在虚拟桌面上。要解决此问题，请删除快捷方式。[#402889]
- 使用 Microsoft Hyper-V 时，如果计算机存储在本地，而虚拟磁盘存储在群集共享卷 (CSV) 上，则可能无法创建具有个人虚拟磁盘的计算机的目录；目录创建失败并出现错误。要解决此问题，请对虚拟磁盘使用备用存储设置。[#423969]
- 首次登录基于 Provisioning Services 目录创建的虚拟桌面时，如果个人虚拟磁盘已重置（使用命令 ctxpvd.exe -s reset），该桌面会提示用户重新启动。要解决此问题，请根据提示重新启动桌面。此为一次性重置，重新登录时不需要再执行此操作。[#340186]
- 如果您在个人虚拟磁盘上安装了 .NET 4.5，而稍后执行映像更新时安装或修改了 .NET 4.0，依赖 .NET 4.5 的应用程序将无法启动。要解决此问题，请分发基础映像中的 .NET 4.5 作为映像更新。
- 另请参阅 XenApp 和 XenDesktop 7.6 版本的
— 已知问题
文档。

关于 Personal vDisk 7.1.3

版本 7.1.3 中修复的 7.1.1 中的已知问题：

- 从 Personal vDisk 5.6.0 直接升级到 Personal vDisk 7.x 可能会导致 Personal vDisk 失败。[#432992]
- 用户有时只能连接到具有个人虚拟磁盘的虚拟桌面。[#437203]
- 如果在将 Personal vDisk 5.6.5 或更高版本升级到 Personal vDisk 7.0 或更高版本的过程中个人虚拟磁盘映像更新操作中断，以后执行更新操作可能会失败。[#436145]

关于 Personal vDisk 7.1.1

版本 7.1.1 中修复的 7.1 中的已知问题：

- 通过映像更新升级到 Symantec Endpoint Protection 12.1.3 会导致 symhelp.exe 报告损坏的防病毒软件定义。[#423429]
- 如果服务控制管理器 (services.exe) 崩溃，Personal vDisk 会导致池桌面重新启动。[#0365351]

版本 7.1.1 中的已知新问题：无

关于 Personal vDisk 7.1

版本 7.1 中的新增功能：

- 现在，可将 Personal vDisk 与运行 Windows 8.1 的桌面结合使用，事件日志记录功能也已得到改进。
- 此版本不再支持写入时复制 (CoW) 功能。Personal vDisk 从版本 7.0 升级 7.1 时，由 CoW 管理的所有数据更改都将丢失。这是 XenDesktop 7 中的实验性功能，默认情况下处于禁用状态，因此如果不启用该功能，将不会有任何影响。

版本 7.1 中修复的 7.0.1 中的已知问题：

- 如果将 Personal vDisk 注册表项 EnableProfileRedirection 的值设置为 1 或 ON，随后在更新映像时，将其更改为 0 或 OFF，则可能将整个个人虚拟磁盘空间分配给用户安装的应用程序，而没有为仍保留在虚拟磁盘上的用户配置文件留出空间。如果为某个目录禁用此配置文件重定向功能，而在映像更新期间将其启用，则用户可能无法登录其虚拟桌面。[#381921]
- Personal vDisk 清单更新失败时，桌面服务不会在事件查看器中记录正确的错误消息。[#383331]
- 升级到 Personal vDisk 7.x 时，修改的规则无法保留下来。对于版本 7.0 到 7.1 的升级，此问题已得到修复。从版本 5.6.5 升级到版本 7.1 时，必须首先保存规则文件，然后在升级后重新应用这些规则。[#388664]
- 运行 Windows 8 的个人虚拟磁盘无法从 Windows 应用商店中安装应用程序。将出现错误消息“您的购买无法完成”。启用 Windows Update Service 无法解决此问题，现在此问题已得到修复。但是，系统重新启动后，必须重新安装用户安装的应用程序。[#361513]
- 在带有个人虚拟磁盘的 Windows 7 池桌面中，某些符号链接丢失。这样，对于将图标存储在 C:\Users\All Users 中应用程序，其图标将不会显示在“开始”菜单中。[#418710]
- 由于在更新清单后对系统做了大量更改，因此，如果更新序列号 (Update Sequence Number, USN) 日志溢出，个人虚拟磁盘可能无法启动。[#369846]
- 状态代码为 0x20 且错误代码为 0x20000028 时，个人虚拟磁盘不会启动。[#393627]
- Symantec Endpoint Protection 12.1.3 显示“Proactive Threat Protection is malfunctioning”（主动威胁保护功能出现故障）消息，并且无法获得此组件的实时更新状态。[#390204]

版本 7.1 中的已知新问题：请参阅 XenDesktop 7.1 版本的

— 已知问题
文档。

关于 Personal vDisk 7.0.1

版本 7.0.1 中的新增功能：Personal vDisk 现在更能适应环境的变化。具有个人虚拟磁盘的虚拟桌面现在注册到 Delivery Controller，即使映像更新失败也是如此，并且不安全地关闭系统不会再将虚拟磁盘置于永久禁用状态。此外，现在还可以使用规则文件在部署期间从虚拟磁盘排除文件和文件夹。

版本 7.0.1 中修复的 5.6.13 中的已知问题：

- 用户在池虚拟桌面上对组成员关系所做的更改在映像更新后可能会丢失。[#286227]
- 映像更新失败并出现磁盘空间不足错误，即使个人虚拟磁盘具有足够的空间也是如此。[#325125]
- 某些应用程序无法安装在具有个人虚拟磁盘的虚拟桌面上，并显示一条消息，指出需要重新启动。这是由一项待执行的重命名操作所致。[#351520]
- 在主映像中创建的符号链接在具有个人虚拟磁盘的虚拟桌面上无效。[#352585]
- 在使用 Citrix Profile Management 和 Personal vDisk 的环境中，如果启用了配置文件重定向，负责检查系统卷上的用户配置文件的程序可能无法正确运行。[#353661]
- 清单大于 2 GB 时，清单更新过程在主映像上失败。[#359768]
- 映像更新失败，错误代码 112，并且个人虚拟磁盘损坏，即使虚拟磁盘具有足够的可用空间来执行更新也是如此。[#363003]
- 对于具有超过 250 个桌面的目录，调整大小脚本失败。[#363365]
- 用户对环境变量所做的更改在执行映像更新时丢失。[#372295]
- 在具有个人虚拟磁盘的虚拟桌面上创建的本地用户在执行映像更新时丢失。[#377964]
- 由于在更新清单后对系统做了大量更改，因此，如果更新序列号 (Update Sequence Number, USN) 日志溢出，个人虚拟磁盘可能无法启动。为避免出现此问题，将主映像中的 USN 日志大小增加到最小 32 MB 并执行映像更新。[#369846]
- 已在个人虚拟磁盘上发现问题，该问题会在取代模式下使用 AppSense 时阻止 AppSense Environment Manager 注册表单元配置操作正常工作。Citrix 和 AppSense 正共同努力解决此问题，该问题与安装 Personal vDisk 时 RegRestoreKey API 的行为有关。[#0353936]

与版本无关的已知问题

- 如果安装在个人虚拟磁盘 (PvD) 上的某个应用程序与另一个安装在主映像上具有相同版本的应用程序相关，在映像更新后，PvD 上的应用程序可能会停止工作。如果卸载主映像上的应用程序或将其升级到更新的版本，则会出现此问题，因为此操作从主映像上删除了 PvD 上的应用程序所需的文件。为防止发生此问题，请在主映像上保留包含 PvD 上的应用程序所需文件的应用程序。
例如，主映像包含 Office 2007，并且用户在 PvD 上安装了 Visio 2007，Office 应用程序和 Visio 运行正常。之后，管理员在主映像上将 Office 2007 替换成 Office 2010，然后使用更新后的映像更新所有受影响的计算机。Visio 2007 将不再可用。为避免此问题，请在主映像上保留 Office 2007。[#320915]
- 如果使用 Personal vDisk，当部署 McAfee Virus Scan Enterprise (VSE) 时，请在主映像上使用版本 8.8 Patch 4 或更高版本。[#303472]
- 如果所创建的关于主映像中某个文件的快捷方式不再起作用（因为快捷方式目标在 PvD 中重命名），请重新创建快捷方式。[#367602]
- 请勿在主映像中使用绝对/硬盘链接。[#368678]
- Personal vDisk 不支持 Windows 7 备份和还原功能。[#360582]
- 应用更新后的主映像后，本地用户和组控制台变得无法访问或显示不一致的数据。要解决此问题，请在 VM 上重置用户帐户，为此需要重置安全配置单元。此问题在 7.1.2 版本中得以修复（适用于在之后的版本中创建的 VM），但是此修复对使用更早的版本创建然后又进行升级的 VM 不起作用。[#488044]
- 在 ESX 虚拟机管理程序环境中使用池 VM 时，如果选择的 SCSI 控制器类型为“VMware Paravirtual”，系统会向用户显示重新启动提示。为解决此问题，请使用 LSI SCSI 控制器类型。[#394039]
- 在通过 Provisioning Services 创建的桌面上重置 PvD 后，用户登录到 VM 后可能会收到重新启动提示。作为一种解决方法，可重新启动桌面。[#340186]
- Windows 8.1 桌面用户可能无法登录其 PvD。管理员可能会看到消息“PvD was disabled due to unsafe shutdown”（由于不安全的关闭操作导致禁用 PvD），并且 PvDActivation 日志可能会包含消息“Failed to load reg hive [\\Device\\IvmVhdDisk00000001\\CitrixPvD\\Settings\\RingCube.dat]”（无法加载 reg 配置单元 [\\Device\\IvmVhdDisk00000001\\CitrixPvD\\Settings\\RingCube.dat]）。当用户的 VM 以不安全的方式关闭时，会出现此问

题。为解决此问题，可重置 Personal vDisk。[#474071]

安装和升级

Nov 02, 2016

Personal vDisk 7.x is supported on XenDesktop version 5.6 through the current version. The "System requirements" documentation for each XenDesktop version lists the supported operating systems for Virtual Delivery Agents (VDAs), and the supported versions of hosts (virtualization resources), and Provisioning Services. For details about Provisioning Services tasks, see the Provisioning Services documentation.

Install and enable PvD

PvD is installed automatically when you install or upgrade a VDA for Desktop OS on a machine. If you update the PvD software after installing the VDA, use the PvD MSI provided [here](#) (Citrix account credentials required).

Enabling PvD:

- If you are using Machine Creation Services (MCS), PvD is enabled automatically when you create a machine catalog of desktop OS machines that will use a personal vDisk.
- If you are using Provisioning Services (PVS), PvD is enabled automatically when you run the inventory during the master (base) image creation process, or when auto-update runs the inventory for you.

VDA installation offers options to enable PvD (by selecting the "Personal vDisk" checkbox in the graphical interface or by specifying the /baseimage option in the command line interface). However, omitting this action during the VDA install (which is the default) still allows you to use the same image to create both PvD desktops and non-PvD desktops, because PvD is enabled during the catalog creation process.

Add personal vDisks

You add personal vDisks to hosts when you configure a Site. You can choose to use the same storage on the host for VMs and personal vDisks, or you can use different storage for personal vDisks.

Later, you can also add personal vDisks and their storage to existing hosts (connections), but not machine catalogs.

1. Select Configuration > Hosting in the Studio navigation pane.
2. Select Add Personal vDisk storage in the Actions pane, and specify the storage location.

Upgrade PvD

The easiest way to upgrade personal vDisk from an earlier 7.x version is to simply upgrade your desktop OS VDAs to the version provided with the most recent XenDesktop version. Then, run the PvD inventory.

You can also upgrade just PvD using the PvD MSI from [here](#).

Uninstall PvD

You can use one of two ways to remove the PvD software:

- Uninstall the VDA; this removes the PvD software as well.
- If you updated PvD using the PvD MSI, then you can uninstall it from the Programs list.

If you uninstall PvD and then want to reinstall the same or a newer version, first back up the registry key HKLM\Software\Citrix\personal vDisk\config, which contains environment configuration settings that might have changed. Then, after installing PvD, reset the registry values that might have changed, by comparing them with the backed-up version.

配置与管理

May 04, 2016

本主题介绍配置和管理 Personal vDisk (PvD) 环境时需要考虑的项目，其中还包括最佳实践指南和任务描述。

对于包括使用 Windows 注册表的过程：

警告：注册表编辑不当会导致严重问题，可能导致需要重新安装操作系统。Citrix 无法保证因“注册表编辑器”使用不当导致出现的问题能够得以解决。使用“注册表编辑器”需自担风险。在编辑注册表之前，请务必进行备份。

注意事项：个人虚拟磁盘大小

以下因素影响主个人虚拟磁盘卷的大小：

- **用户将在其 PvD 上安装的应用程序的大小**

重新启动时，PvD 会确定应用程序区域 (UserData.v2.vhd) 中剩余的可用空间。如果可用空间低于 10%，应用程序区域将扩展到任何未使用的配置文件区域空间中（默认为驱动器 P: 上的可用空间）。添加到应用程序区域的空间大约占应用程序区域和配置文件区域中剩余的总可用空间的 50%。

例如，如果 10 GB PvD 上的应用程序空间（默认大小为 5 GB）达到 4.7 GB，配置文件区域具有 3 GB 可用空间，则添加到应用程序区域的新增空间的计算方法如下：

增加的空间 = $(5.0 - 4.7) / 2 + 3.0 / 2 = 1.65$ GB

添加到应用程序区域的空间只是一个大约值，因为会提供小额的补贴空间来存储日志以及用于开销。计算及可能的调整大小操作在每次重新启动时执行。

- **用户配置文件的大小（如果未使用独立的配置文件管理解决方案）**

除应用程序所需的空间外，还应确保个人虚拟磁盘上有足够的可用空间来存储用户的配置文件。计算空间要求时请包括任何非重定向特殊文件夹（如“我的文档”和“我的音乐”）。可以从“控制面板”(sysdm.cpl) 获取现有配置文件的大小。

某些配置文件重定向解决方案存储根文件（标记文件）而非真实的配置文件数据。这些配置文件解决方案可能显示为最初未存储任何数据，但实际上每个存根文件占用文件系统中的文件目录条目；通常情况下，此数量大约为每个文件 4 KB。如果您采用此类解决方案，必须根据实际的配置文件数据（而非存根文件）来预估大小。

企业文件共享应用程序（如 ShareFile 和 Dropbox）可能会将数据同步或下载到个人虚拟磁盘上的用户配置文件区域。如果您采用此类解决方案，预估大小时必须包括足够的空间来存储此数据。

- **包含 PvD 清单的模板 VHD 占用的开销**

模板 VHD 包含 PvD 清单数据（与主映像内容对应的标记文件）。PvD 应用程序区域将基于此 VHD 创建。由于每个 sentinel 文件或文件夹由文件系统中的文件目录条目组成，因此，模板 VHD 内容将占用 PvD 应用程序空间，即使在最终用户安装任何应用程序之前也是如此。可以通过在创建清单后浏览主映像来确定模板 VHD 的大小。或者，也可以使用以下等式估算其大小：

模板 VHD 大小 = (基础映像上的文件数) × 4 KB

可以通过在基础 VM 映像中的驱动器 C: 上单击鼠标右键并选择属性来确定文件和文件夹的数量。例如，包含 25 万个文件的映像的模板 VHD 大小约为 1,024,000,000 字节（接近 1 GB）。此空间不可用于 PvD 应用程序区域中的应用程序安装。

- **PvD 映像更新操作的开销**

执行 PvD 映像更新操作期间，PvD 的根目录处（默认为 P:）必须有足够的可用空间，以便合并来自两个映像版本的变更以及用户对其 PvD 所做的更改。通常情况下，PvD 会预留几百 MB 空间用于此目的，但写入到 P: 驱动器的额外数据可能会占

用这些预留的空间，致使没有足够的空间来成功完成映像更新。PvD 池统计数据脚本（位于 XenDesktop 安装介质上的 Support/Tools/Scripts 文件夹中）或 PvD 映像更新监视工具（位于 Support/Tools/Scripts\PvdTool 文件夹中）可以帮助识别目录中正在更新以及接近满载的 PvD 磁盘。

如果存在防病毒产品，可能会影响运行清单或执行更新所需的时间。如果将 CtxPvD.exe 和 CtxPvDSvc.exe 添加到防病毒产品的排除列表，则可以提高性能。这些文件位于 C:\Program Files\Citrix\personal vDisk\bin 中。从防病毒软件执行的扫描中排除这些可执行文件最高可将清单和映像更新的性能提升十倍。

- **非预期增长（非预期应用程序安装等）的开销**

考虑在总大小的基础上额外留出一定的空间（固定量或虚拟磁盘大小的百分比），用于用户在部署期间执行的任何非预期应用程序安装。

方法：配置个人虚拟磁盘大小和分配

可以通过设置 VHD 的初始大小手动调整用于决定相对于驱动器 P: 的 VHD 大小的自动调整大小算法。例如，如果您知晓用户将安装大量应用程序，但这些应用程序过大，无法安装在 VHD 上，即使通过算法调整大小之后也是如此，则此操作将非常有用。在此情况下，可以增加应用程序空间的初始大小，以容纳用户安装的应用程序。

更可取的做法是调整主映像上的 VHD 的初始大小。或者，当用户没有足够的空间来安装应用程序时，也可以调整虚拟桌面上 VHD 的大小。但是，必须在每个受影响的虚拟桌面上分别重复此操作；无法调整已创建目录中的 VHD 初始大小。

请确保 VHD 足够大，能够存储防病毒定义文件，这些文件通常都非常大。

查找并设置 HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\personal vDisk\Config 中的注册表项。（请勿修改此注册表项中的其他设置。）所有设置必须在主映像上指定（MinimumVHDSIZEInMB 除外，可以在单个计算机上更改此设置）；在主映像上指定的设置在下次映像更新时应用。

- **MinimumVHDSIZEInMB**

指定个人虚拟磁盘的应用程序部分 (C:) 的最低大小（以 MB 为单位）。新大小必须大于现有大小，但小于磁盘大小减去 PvDReservedSpaceMB 所得的值。

增大此值会将虚拟磁盘上配置文件部分中的可用空间分配给 C:。如果使用的值小于当前 C: 驱动器的大小，或者如果 EnableDynamicResizeOfAppContainer 设置为 0，则将忽略此设置。

默认值：2048

- **EnableDynamicResizeOfAppContainer**

启用或禁用动态调整大小算法。

- 设置为 1 时，当 C: 上的可用空间下降到 10% 以下时，应用程序空间（位于 C: 上）将自动调整大小。允许使用的值为 1 和 0。需要重新启动才能使调整的大小生效。
- 设置为 0 时，将根据 7.x 之前的 XenDesktop 版本中使用的方法确定 VHD 的大小。

默认值：1

- **EnableUserProfileRedirection**

启用或禁用将用户的配置文件重定向到虚拟磁盘。

- 设置为 1 时，PvD 将用户的配置文件重定向到个人虚拟磁盘驱动器（默认情况下为 P:）。配置文件通常重定向到 P:\Users，与标准 Windows 配置文件相对应。此重定向会保留配置文件，以防需要重置 PvD 桌面。
- 如果设置为 0，虚拟磁盘上的所有空间减去 PvDReservedSpaceMB 所得的值将分配给 C:，即虚拟磁盘应用程序部分，并且虚拟磁盘驱动器 (P:) 在 Windows 资源管理器中处于隐藏状态。Citrix 建议在使用 Citrix Profile Management 或其他漫游配置文件解决方案时通过将此设置更改为 0 来禁用重定向。

此设置会将配置文件保留在 C:\Users 中（而非将其重定向到虚拟磁盘），并使漫游配置文件解决方案能够处理配置文

件。

此值可确保将 P: 上的所有空间都分配给应用程序。

本主题做以下假设，即将此值设置为 0 时，配置文件管理解决方案已准备就绪。如果漫游配置文件解决方案未准备就绪，则建议不要禁用配置文件重定向，因为后续的 PvD 重置操作会导致删除配置文件。

更新映像时请勿更改此设置，因为此设置不会更改现有配置文件的位置，但会将个人虚拟磁盘上的所有空间分配给 C: 并隐藏 PvD。

请在部署目录前配置此值。部署目录后无法更改此值。

重要：自 XenDesktop 7.1 以来，执行映像更新时不会保留对此值所做的更改。在第一次创建配置文件的来源目录时设置密钥值。以后将无法修改重定向行为。

默认值：1

- **PercentOfPvDForApps**

设置虚拟磁盘的应用程序部分 (C:) 与配置文件部分之间的拆分。如果将 EnableDynamicResizeOfAppContainer 设置为 0，则创建新 VM 时以及执行映像更新期间会使用此值。

仅当将 EnableDynamicResizeOfAppContainer 设置为 0 时，更改 PercentOfPvDForApps 设置才会产生差别。默认情况下，将 EnableDynamicResizeOfAppContainer 设置为 1（启用），这意味着仅当 AppContainer（您看到的是 C 驱动器）接近满载时，也就是可用空间低于 10% 时，才得以扩展（即动态扩展）。

增大 PercentOfPvDForApps 时仅增大允许 Apps 部分扩展到的最大空间。该操作不会立即置备空间。您还必须在主映像中配置拆分分配，所配置的设置将在下次映像更新时应用。

如果已在将 EnableDynamicResizeOfAppContainer 设置为 1 的情况下生成计算机目录，请在主映像中将此设置更改为 0，以便下次更新时应用，并配置一个恰当的分配拆分。只要请求的拆分大小大于当前为 C 驱动器分配的大小，则始终应用该大小。

如果您希望保持对空间拆分的完整控制，应将此值设置为 0。这允许您完整控制 C 驱动器的大小，而不依赖占用的空间低于阈值的用户扩展该驱动器。

默认值：50%（为两部分分配相等的空间）

- **PvDReservedSpaceMB**

指定虚拟磁盘上为存储 Personal vDisk 日志及其他数据而保留的空间大小（以 MB 为单位）。

如果您的部署包含 XenApp 6.5（或更早的版本）并使用应用程序流技术推送，请根据 Rade 缓存的大小增大此值。

默认值：512

- **PvDResetUserGroup**

仅适用于 XenDesktop 5.6 - 允许指定的用户组重置个人虚拟磁盘。之后的 XenDesktop 版本使用委派管理员实现此目的。

其他设置：

- **Windows Update 服务** - 确保在主映像中将 Windows 更新设置为从不检查更新，将 Windows Update 服务设置为已禁用。如果需要在 PvD 上运行 Windows Update 服务，将其设置为从不检查更新有助于阻止在关联的计算机上安装更新。Windows 8 应用商店需要运行此服务才能安装现代风格的应用程序。
- **Windows 更新** - 包括 Internet Explorer 更新，必须在主映像上应用。

- **更新要求重新启动** - 应用到主映像的 Windows 更新可能要求多次重新启动才能完全安装，具体取决于这些更新提供的修补程序类型。请务必先正确重新启动主映像以全面完成应用到该映像的任何 Windows 更新的安装，然后再创建 PvD 清单。
- **应用程序更新** - 更新主映像上安装的应用程序以节省用户虚拟磁盘上的空间。此设置还可避免重复更新每个用户的虚拟磁盘上的应用程序。

注意事项：主映像上的应用程序

某个软件可能与 PvD 组成用户环境的方式相冲突，因此，必须将其安装在主映像上（而不是单个计算机上）以避免这些冲突。此外，尽管某些其他软件并不会与 PvD 操作冲突，Citrix 还是建议将其安装在主映像上。

必须安装在主映像上的应用程序：

- 代理和客户端（例如，System Center Configuration Manager Agent、App-V 客户端、Citrix Receiver）
- 用于安装或修改早期启动驱动程序的应用程序
- 用于安装打印机或扫描仪软件/驱动程序的应用程序
- 用于修改 Windows 网络堆栈的应用程序
- VMware Tools 和 XenServer Tools 等 VM 工具

应该安装在主映像上的应用程序：

- 分发给大量用户的应用程序。在每种情况下，都请在部署之前关闭应用程序更新：
 - 使用批量许可的企业应用程序，例如 Microsoft Office、Microsoft SQL Server
 - 常见应用程序，例如 Adobe Reader、Firefox 和 Chrome
- 大型应用程序（例如 SQL Server、Visual Studio）和应用程序框架（例如 .NET）

以下建议和限制适用于用户在具有个人虚拟磁盘的桌面上安装的应用程序。如果用户具有管理特权，则不能强制执行其中某些建议和限制：

- 用户不应卸载主映像上的应用程序并在其个人虚拟磁盘上重新安装相同的应用程序。
- 更新或卸载主映像上的应用程序时应小心谨慎。在映像上安装某个版本的应用程序时，用户可能会安装一个需要此版本的加载项应用程序（例如插件）。如果存在此类依赖项，则更新或卸载该映像上的应用程序可能会导致加载项无法正常使用。例如，在主映像上安装 Microsoft Office 2010 后，用户在其个人虚拟磁盘中安装 Visio 2010。以后升级主映像上的 Office 可能会导致本地安装的 Visio 不可用。
- 不支持具有依赖于硬件的许可证的软件（通过硬件保护装置或基于签名的硬件）。

注意事项：Provisioning Services

同时使用 Provisioning Services 和 PvD 时：

- 必须将 Soap Service 帐户添加到 Citrix Studio 的“管理员”节点，并且必须具有“计算机管理员”或权限更高的角色。这样可确保在将 Provisioning Services (PVS) 虚拟磁盘提升到生产模式时将 PvD 桌面置于“正在准备”状态。
- 必须使用 Provisioning Service 版本控制功能来更新 Personal vDisk。将版本提升到生产模式时，Soap Service 会将 PvD 桌面置于“正在准备”状态。
- 个人虚拟磁盘的大小应始终大于 Provisioning Services 写入缓存磁盘的大小（否则，Provisioning Services 可能错误地选择个人虚拟磁盘用作其写入缓存）。
- 创建交付组后，可以使用 [PvD 映像更新监视工具](#) 或 [resize](#) 和 [poolstats](#) 脚本 (personal-vdisk-poolstats.ps1) 监视个人虚拟磁盘。

正确设置写入缓存磁盘的大小。常规操作期间，PvD 捕获大多数用户写入（更改）并将其重定向到个人虚拟磁盘。这表示您可以降低 Provisioning Services 写入缓存磁盘的大小。但是，当 PvD 不活动时（例如执行映像更新操作期间），小型 Provisioning Services 写入缓存磁盘可能会填满数据，导致计算机崩溃。

Citrix 建议根据 Provisioning Services 最佳实践来确定 Provisioning Services 写入缓存磁盘的大小，并且添加大小等于主映像上的模板 VHD 大小两倍的额外空间。合并操作需要所有这些空间的可能性非常低，但仍存在这种可能性。

使用 Provisioning Services 置备包含启用 PvD 的计算机的目录时：

- 请遵循 Provisioning Services 文档中的指南。
- 可以通过编辑 Studio 中的连接，更改电源操作限制设置；请参见下文。
- 如果更新 Provisioning Services 虚拟磁盘，请在安装/更新应用程序和其他软件并重新启动虚拟磁盘后，运行 PvD 清单，然后关闭 VM。然后，将新版本提升到生产模式。目录中的 PvD 桌面应自动进入“正在准备”状态。如果没有进入此状态，请检查 Soap Service 帐户在 Controller 上是否具有计算机管理员权限或更高权限。

利用 Provisioning Services 测试模式功能，您可以创建包含使用已更新主映像的计算机的目录。如果测试确认了测试目录的可行性，则可以将其提升为生产模式。

注意事项：Machine Creation Services

使用 Machine Creation Services (MCS) 部署包含启用 PvD 的计算机的目录时：

- 请遵循 XenDesktop 文档中的指南。
- 创建主映像后运行 PvD 清单，然后关闭 VM（如果不关闭 VM，PvD 无法正常运行。）然后，创建主映像的快照。
- 在“创建计算机目录”向导中，指定个人虚拟磁盘大小和驱动器盘符。
- 创建交付组后，可以使用 [PvD 映像更新监视工具](#) 或 [resize](#) 和 [poolstats](#) 脚本 (personal-vdisk-poolstats.ps1) 监视个人虚拟磁盘。
- 可以通过编辑 Studio 中的连接，更改电源操作限制设置；请参见下文。
- 如果您更新主映像，请在更新映像上的应用程序和其他软件后运行 PvD 清单，然后关闭 VM。然后，创建主映像的快照。
- 使用 PvD 映像更新监视工具或 personal-vdisk-poolstats.ps1 脚本验证将使用更新后的主映像且启用 PvD 的每个 VM 上是否有足够的空间。
- 更新计算机目录后，PvD 桌面进入“正在准备”状态，因为它们各自分别处理新主映像中的更改。这些桌面根据在计算机更新过程中指定的推行策略进行更新。
- 使用 PvD 映像更新监视工具或 personal-vdisk-poolstats.ps1 脚本监视处于“正在准备”状态的 PvD。

方法：排除虚拟磁盘中的文件和文件夹

使用规则文件从虚拟磁盘排除文件和文件夹。可以在部署个人虚拟磁盘期间执行此项操作。规则文件按 custom_*_rules.template.txt 格式命名并存储在 \config 文件夹中。各个文件中的注释提供了其他文档。

方法：更新主映像时运行清单

如果启用 PvD 并在安装后对主映像进行了更新，之后应刷新磁盘的清单（称为“运行清单”）并创建新快照，这一点很重要。

由于管理员（而非用户）管理主映像，因此，如果您安装的某个应用程序将二进制文件放在管理员的用户配置文件中，则共享虚拟桌面（包括基于池计算机目录以及使用 PvD 的池计算机目录的共享虚拟桌面）的用户将无法使用此应用程序。用户必须自己安装此类应用程序。

完成下述过程中的每个步骤后，最好生成一张映像快照。

1. 通过执行以下操作升级主映像：安装任何应用程序或操作系统更新，然后在计算机上执行任何系统配置操作。
对于要通过 Personal vDisk 部署的 Windows XP 为基础的主映像，请确认未打开任何对话框（例如，确认软件安装情况的消息或使用未签名驱动程序的提示）。此环境中主映像上打开的对话框将阻止 VDA 注册到 Delivery Controller。可以使用“控制面板”防止显示使用未签名应用程序的提示。例如，导航到“系统”>“硬件”>“驱动程序签名”，然后选择忽略警告对应的选项。
2. 关闭计算机。对于 Windows 7 计算机，请在 Citrix Personal vDisk 阻止关机时单击取消。
3. 在 Citrix Personal vDisk 对话框中，单击更新清单。此步骤可能需要几分钟时间才能完成。
重要：如果您中断了随后的关机过程（即使是对映像执行少量更新），Personal vDisk 的清单也不再与主映像一致。这将导致 Personal vDisk 功能停止运行。如果您中断了关闭过程，则必须重新启动计算机，将其关闭，然后在系统提示时再次单击

更新清单。

4. 清单操作关闭计算机后，请生成一张主映像快照。

可以将清单导出到网络共享，然后再将此清单导入到主映像。有关详细信息，请参阅[导出和导入 PvD 清单](#)。

方法：配置连接限制设置

Citrix Broker Service 控制提供桌面和应用程序的计算机的电源状态。Broker Service 可以通过一个 Delivery Controller 控制多个虚拟机管理程序。Broker 电源操作控制 Controller 与虚拟机管理程序之间的交互。为避免虚拟机管理程序过载，向更改计算机电源状态的操作分配优先级，并使用限制机制将其发送到虚拟管理程序。以下设置影响限制。可以通过在 Studio 中编辑连接（“高级”页面）指定这些值。

要配置连接限制值，请执行以下操作：

1. 在 Studio 导航窗格中，选择配置 > 托管。
2. 选择连接，然后在“操作”窗格中选择编辑连接。
3. 可以更改以下值：
 - **同步操作(所有类型)** - 允许同时进行的电源操作的最大数。此设置同时指定虚拟机管理程序连接的绝对值和百分比。使用两个值中的较小值。
默认值：绝对值 100，20%
 - **同步个人虚拟存储清单更新** - 允许并发的个人虚拟磁盘电源操作的最大数。此设置同时指定连接的绝对值和百分比。使用两个值中的较小值。
默认值：绝对值 50，25%

计算绝对值：确定最终用户存储支持的 IOPS 总数 (TIOPS) (此值由制造商指定或通过计算机得出)。每个 VM 使用 350 次 IOPS (IOPS/VM)，确定在给定时间存储上应该活动的 VM 数。通过使用 IOPS 总数除以 IOPS/VM 计算此值。

例如，如果最终用户存储为 14000 IPS，则活动 VM 数为 $14000 \text{ IOPS} / 350 \text{ IOPS/VM} = 40$ 。

- **每分钟最大新操作数** - 每分钟可以发送给虚拟机管理程序的最大新电源操作数。以绝对值形式提供。
默认值：10

为帮助找出部署中有关这些设置的最佳值，请遵循以下说明：

1. 使用默认值测量测试目录的映像更新所需的总响应时间。此值是指映像更新开始时间 (T1) 与目录中的最后一个计算机上的 VDA 向 Controller 注册的时间 (T2) 之间的差。总响应时间 = $T2 - T1$ 。
2. 测量映像更新期间虚拟机管理程序存储的每秒钟输入/输出操作数 (IOPS)。此数据可用作优化基准。（默认值可能是最佳设置；但是，系统也可能会达到最大 IOPS，此时需要降低设置值。）
3. 请按照下面的描述降低“同步个人虚拟存储清单更新”值（保持其他所有设置不变）。
 1. 将此值增加 10，并在每次更改后测量总响应时间。继续将此值增加 10 并测试结果，直到总响应时间降低或无变化。
 2. 如果前一步的结果显示通过增加值未得到改善，则以 10 为增量降低此值，并在每次降低后测量总响应时间。重复此步骤，直到总响应时间保持不变或未得到进一步改善。此值很可能就是最佳的 PvD 电源操作值。
4. 获得 PvD 电源操作设置值后，调整“同步操作(所有类型)”值和“每分钟最大新操作数”值，每次调整一个。按照以上所述过程（按增量增加或降低）测试其他值。

方法：具有 PvD 的 System Center Configuration Manager 2007

System Center Configuration Manager (Configuration Manager) 2012 无需任何特殊配置，可以像安装其他主映像应用程序那样安装。下列信息仅适用于 System Center Configuration Manager 2007。不支持 Configuration Manager 2007 之前的 Configuration Manager 版本。

要在 PvD 环境中使用 Configuration Manager 2007 代理软件，请完成以下步骤。

1. 在主映像上安装客户端代理。
 1. 在主映像上安装 Configuration Manager 客户端。
 2. 停止并禁用 ccmexec 服务（SMS 代理）。
 3. 按如下所示从本地计算机证书存储中删除 SMS 或客户端证书：
 - 混合模式：证书(本地计算机)\SMS\证书
 - 本机模式
 - 证书(本地计算机)\个人\证书
 - 删除证书颁发机构颁发的客户端证书（通常为内部公钥基础结构）
 4. 删除或重命名 C:\Windows\smscfg.ini。
2. 删除唯一标识客户端的信息。
 1. （可选）删除或移动 C:\Windows\System32\CCM\Logs 中的日志文件。
 2. 安装 Virtual Delivery Agent（如果以前未安装）并创建 PvD 清单。
 3. 关闭主映像，创建快照，然后使用此快照创建计算机目录。
3. 验证 Personal vDisk 并启动服务。在每个 PvD 桌面首次启动后执行一次这些步骤。例如，可以使用域 GPO 完成此操作。
 - 通过检查是否存在注册表项 HKLM\Software\Citrix\personal vDisk\config\virtual 来确认 PvD 处于活动状态。
 - 将 ccmexec 服务（SMS 代理）设置为“自动”并启动该服务。Configuration Manager 客户端与 Configuration Manager 服务器联系，并检索新的唯一证书和 GUID。

工具

May 04, 2016

可以使用以下工具和实用程序自定义、加快和监视 PVD 操作。

自定义规则文件

利用 PVD 提供的自定义规则文件，可以采用以下方式修改 PVD 映像更新的默认行为：

- PVD 上的文件的可见性
- 如何合并对文件所做的更改
- 文件是否可写入

有关自定义规则文件及 CoW 功能的详细说明，请参阅位于以下位置的文件中的注释：安装了 PVD 的 VM 上的 C:\ProgramData\Citrix\personal vDisk\Config。名为 custom_*. 的文件介绍了相关规则及其启用方法。

resize 和 poolstats 脚本

提供了两个用于监视和管理 PVD 大小的脚本，位于 XenDesktop 安装介质中的 Support\Tools\Scripts 文件夹中。还可以使用 PVD 映像更新监视工具，位于 Support\Tools\Scripts\PvdTool 文件夹中；有关详细信息，请参阅 <http://blogs.citrix.com/2014/06/02/introducing-the-pvd-image-update-monitoring-tool/>。

使用 resize-personalvdisk-pool.ps1 可增大某个目录的所有桌面中的 PVD 的大小。必须在运行 Studio 的计算机上为您的虚拟机管理程序安装以下管理单元或模块：

- XenServer 需要 XenServerPSSnapin
- vCenter 需要 vSphere PowerCLI
- System Center Virtual Machine Manager 需要 VMM 控制台

使用 personal-vdisk-poolstats.ps1 可检查映像更新的状态以及一组 PVD 中的应用程序和用户配置文件的空间。在更新映像之前运行此脚本可检查任何桌面的空间是否已不足，有助于防止更新失败。此脚本需要在 PVD 桌面上启用 Windows Management Instrumentation (WMI-In) 防火墙。可以在主映像上或通过 GPO 启用该防火墙。

如果映像更新失败，则“Update”（更新）列中的条目将指出原因。

重置应用程序区域

如果桌面损坏（由于安装损坏的应用程序或某些其他原因所致），可以将 PVD 的应用程序区域恢复到出厂默认（空）状态。重置操作会使用户配置文件数据保持不变。

要重置 PVD 的应用程序区域，请使用以下方法之一：

- 以管理员身份登录用户的桌面。启动命令提示窗口并运行命令 C:\Program Files\Citrix\Personal vDisk\bin\CtxPVD.exe -s Reset。
- 在 Citrix Director 中找到用户的桌面。单击重置个人虚拟磁盘，然后单击确定。

导出和导入 PVD 清单

映像更新过程是将新映像推向 PVD 桌面的不可或缺部分，其中包括调整现有个人虚拟磁盘以用于新基础映像。对于使用 Machine Creations Services (MCS) 的部署，可以将清单从活动 VM 导出到网络共享，然后再将其导入到主映像中。在主映像中使用此清单计算差额。尽管不强制使用导出/导入清单功能，但是此功能可以改善整个映像更新过程的性能。

要使用导出/导入清单功能，您必须是管理员。如需要，请使用“net use”对用于导出/导入的文件共享进行身份验证。用户上下文必须可以访问用于导出/导入的任何文件共享。

- 要导出清单，请在包含 VDA（版本最低为 7.6）且启用 PVD 的计算机上以管理员身份运行导出命令。
Ctxpvdsvc.exe exportinventory “<导出位置的路径>”

软件会检测当前清单的位置，并将清单导出到指定位置上名为“ExportedPvdInventory”的文件夹中。下面是命令输出摘录：

```
C:\Program Files\Citrix\personal vDisk\bin> .\CtxPvDSvc.exe exportinventory \\share location\ExportedInventory Current inventory source location C:\CitrixPVD\Settings\Inventory\VER-LAS ... Exporting
```

- 要导入之前导出的清单，请在主映像上以管理员身份运行导入命令：

导入

在主映像上以管理员身份运行导入命令。

```
Ctxpvdsvc.exe importinventory “<导出清单的路径>”
```

<导出清单的路径> 应为清单文件的完整路径，通常为 <网络位置\ExportedPvdInventory>。

从导入位置获取清单（之前使用 exportinventory 选项导出清单的位置）并将清单导入到主映像上的清单存储。下面是命令输出摘录：

```
C:\Program Files\Citrix\personal vDisk\bin> .\CtxPvDSvc.exe importinventory \\share location\ExportedInventory\ExportedPvdInventory Importing inventory \\share location\ExportedInventory\ExportedPvdInventory
```

导出后，网络共享应包含以下文件名。导入后，主映像上的清单存储中应包含相同的文件名。

- 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

备份和还原

重要：这两个脚本不会将 PVD 移至新存储位置。您必须通过另外某种方法执行该操作。

可以利用产品安装介质上提供的两个 PowerShell 脚本（在 Support\Tools\Scripts 文件夹中）来备份和还原个人虚拟磁盘。使用备份和还原脚本可将现有 PVD 和用户关联从一个目录迁移到其他目录。如果要更改 PVD 存储，则运行备份和还原脚本将非常有用。备份脚本会创建一个包含现有目录中的元数据的 .xml 文件。该元数据中包含存储上的 PVD 的当前位置以及与 PVD 的用户关联。还原脚本使用该 .xml 文件将 PVD 与新目录关联，并将正确的用户分配给 PVD。

- migration-backup.ps1 可以捕获每个用户与其计算机目录中个人虚拟磁盘之间的映射，并将此信息存储在 .xml 文件中
- migration-restore.ps1 使用该 .xml 文件在计算机目录中重新创建用户桌面

在执行备份和还原之前，请注意以下问题：

- 这些脚本使用虚拟机管理程序 API，因此，必须在执行脚本的 Controller 上安装虚拟机管理程序的 PowerShell 管理单元
- 从一个有权访问用来创建计算机目录的 Controller 的位置运行脚本
- 以下虚拟机管理程序平台支持这些脚本：Citrix XenServer、Microsoft Hyper-V 和 VMware ESX

备份计算机目录

在更改计算机目录时执行备份。可以在计算机目录中的计算机处于活动状态时执行备份。

使用 migration-backup.ps1 备份包含个人虚拟磁盘的任何计算机目录。此脚本将要求提供计算机目录的名称以及虚拟机管理程序的连接信息。然后对计算机目录中所有用户分配的计算机进行迭代，并且将存储每个计算机的个人虚拟磁盘存储和已分配用户之间的映射。此信息位于 .xml 文件中，此文件具有如下结构：

- PvdMigration.hypervisor.Type 支持 VMware ESX、Citrix XenServer 和 Microsoft Hyper-V。
- PvdMigration.PVD 存储有关个人虚拟磁盘的存储位置和与其关联的用户的信息。
- PvdMigration.PVD.DiskId 是执行备份时所在的虚拟机管理程序上的虚拟磁盘的唯一标识符。
- PvdMigration.PVD.DiskName 是 .vhd 或 .vmdk 文件的名称。
- PvdMigration.PVD.SRName 是执行备份时存储提供程序的名称。
- PvdMigration.PVD.SRID 是执行备份时所在的虚拟机管理程序上的存储提供程序的唯一标识符。
- PvdMigration.PVD.UserName 是与此虚拟磁盘关联的用户的名称。
- PvdMigration.PVD.UserSid 是与此虚拟磁盘关联的用户的 SID。
- PvdMigration.PVD.State 指示此虚拟磁盘的状态。此状态可以为“已备份”或“已处理”。在初始备份之后，此状态为“已备份”；在使用 .xml 文件从备份中进行还原之后，此状态将变为“已处理”。

还原计算机目录

在执行还原之前，请注意以下问题：

- 只能还原与备份的计算机目录共享同一主映像的计算机目录
- 必须通过更新创建备份的计算机目录所在的主映像清单来创建新的主映像

使用 migration-restore.ps1 可还原包含个人虚拟磁盘的任何计算机目录。此脚本所需的输入如下：

- 在备份过程中创建的 .xml 文件
- 要还原的计算机目录的名称
- 未连接的个人虚拟磁盘的存储位置的名称。此信息已在 .xml 文件中列出
- 虚拟机管理程序连接信息

migration-restore.ps1 脚本可以找出计算机目录中所有未分配的计算机，并将用户分配给这些计算机。另外还将用户的个人虚拟磁盘连接到这些计算机。

示例方案 1：使用新的计算机名称还原计算机目录及其个人虚拟磁盘

在此方案中，将还原整个计算机目录以及其中连接到计算机的个人虚拟磁盘。将为这些计算机提供新名称。虚拟机管理程序或存储主机发生故障时或者将用户迁移到新的基础结构时，可能需要执行此方案。

1. 运行 migration-backup.ps1 捕获 .xml 文件中的用户到个人虚拟磁盘的映射。
2. 使用备份解决方案将原始计算机目录中的个人虚拟磁盘移动或捕获至磁盘：
 - VMware ESX 或 Microsoft Hyper-V：个人虚拟磁盘位于 Controller 指定的存储上，该存储位于包含虚拟磁盘所连接计算机的名称的文件夹中。
 - Citrix XenServer：个人虚拟磁盘位于 Controller 指定的存储的根目录中。每个虚拟磁盘的名称为 GUID。
3. 使用存储备份解决方案从原始计算机目录中还原个人虚拟磁盘：
 - ESX 或 Hyper-V：将虚拟磁盘置于新存储资源的新文件夹中。或者，将虚拟磁盘保留在新存储资源上的原始路径中。
 - XenServer：将虚拟磁盘置于新存储资源的根目录中。
4. 从用于创建失败的计算机目录的主映像中创建 Provisioning Services 虚拟磁盘或 Machine Creation Services 快照。
5. 从虚拟磁盘或快照的“开始”菜单中运行更新清单。
6. 使用与失败的（原始）计算机目录不同的命名约定在 Studio 中重新创建计算机目录。此操作将生成新计算机的目录，每台计算机均具有站点数据库可识别的新个人虚拟磁盘。
7. 确认已将重新创建的计算机目录分配给正确的交付组。
8. 确认交付组处于维护模式且其中的计算机已关闭。
9. 编辑备份脚本生成的 .xml 文件：
 - ESX 或 Hyper-V：如果在步骤 3 中将虚拟磁盘还原到新存储资源上的新文件夹，对于该文件中的每个 PVD 部分，请将 DiskName 中的文件夹名称替换为还原的虚拟磁盘的位置。如果将虚拟磁盘还原到新存储上的原始路径，请跳过此步骤。
 - XenServer：请跳过此步骤。
10. 在 Controller 上，运行 migration-restore.ps1，并指定 .xml 文件的名称以及存储备份虚拟磁盘的位置。

示例方案 2：重复使用现有计算机名称还原计算机目录及其个人虚拟磁盘

在此方案中，将还原整个计算机目录以及其中连接到计算机的个人虚拟磁盘。将重复使用现有（失败的）计算机名称。虚拟机管理程序或存储主机发生故障时，可能需要执行此方案。

1. 运行 migration-backup.ps1 捕获用户到个人虚拟磁盘的映射。
2. 使用备份解决方案将原始计算机目录中的个人虚拟磁盘移动或捕获至磁盘：
 - ESX 或 Hyper-V：个人虚拟磁盘位于 Controller 指定的存储上，该存储位于包含虚拟磁盘所连接计算机的名称的文件夹中。
 - XenServer：个人虚拟磁盘位于 Controller 指定的存储的根目录中。每个虚拟磁盘的名称为 GUID。
3. 使用存储备份解决方案从原始计算机目录中还原个人虚拟磁盘：
 - ESX 或 Hyper-V：将虚拟磁盘置于新存储资源的新文件夹中。
 - XenServer：将虚拟磁盘置于新存储资源的根目录中。
4. 从用于创建失败的计算机目录的主映像中创建 Provisioning Services 虚拟磁盘或 Machine Creation Services 快照。
5. 从虚拟磁盘或快照的“开始”菜单中运行更新清单。
6. 使用与失败的计算机目录相同的命名约定在 Studio 中重新创建计算机目录。此操作将生成新计算机的目录，每台计算机均具有站点数据库可识别的新个人虚拟磁盘。
7. 确认已将重新创建的计算机目录分配给正确的交付组。
8. 确认桌面组处于维护模式且其中的计算机已关闭。
9. 编辑备份脚本生成的 .xml 文件：
 - ESX 或 Hyper-V：对于该文件中的每个 PVD 部分，请将 DiskName 中的文件夹名称替换为还原的虚拟磁盘的位置。
 - XenServer：请跳过此步骤。

10. 在 Controller 上运行 migration-restore.ps1 脚本，并使用修改的 .xml 文件作为输入。该脚本无需移动虚拟磁盘便可对其进行连接。
11. 确认已成功还原用户的数据。

示例方案 3：在计算机目录中还原个人虚拟磁盘的子集。

在此方案中，计算机目录中的一些但并非全部个人虚拟磁盘发生故障并得到还原。目录中的虚拟机未发生故障。

1. 运行 migration-backup.ps1 捕获 .xml 文件中的用户到个人虚拟磁盘的映射。
2. 在该 .xml 文件中，针对计算机目录中的每个用户均具有 PVD 部分。对于无需还原个人虚拟磁盘的所有用户，请从该文件中删除这些用户及其关联的部分。
3. 使用备份解决方案从原始计算机目录中还原个人虚拟磁盘，如其他方案之一中所述：
 - 要使用新的计算机名称，请采用示例方案 1。
 - 要保留计算机名称，请采用示例方案 2。
4. 确保目录中存在足够的未分配计算机。根据需要添加计算机。每个要还原虚拟磁盘的用户均需要新的计算机。
5. 确认桌面组处于维护模式且其中的计算机已关闭。
6. 在 Controller 上运行 migration-restore.ps1，并使用修改的 .xml 文件作为输入。
7. 确认已成功还原用户的数据。

显示、消息和故障排除

May 04, 2016

在 Studio 中，选择计算机目录中启用 PvD 的计算机时，“PvD”选项卡将提供映像更新期间的监视状态，以及预计完成时间和进度。在映像更新期间可能显示的状态为：就绪、正在准备、正在等待、失败和已请求。

映像更新失败可能由多种不同的原因，包括空间不足，或者桌面在足够的时间内未找到 PvD。Studio 指示映像更新失败时，会提供错误代码和描述性文本，以帮助进行故障排除。使用 Personal vDisk 映像更新监视工具或 personal-vdisk-poolstats.ps1 脚本监视映像更新进度并获取与失败有关的错误代码。

如果映像更新失败，以下日志文件可提供进一步的故障排除信息：

- PvD 服务日志 - C:\ProgramData\Citrix\personal vDisk\Logs\PvDsvc.log.txt
- PvD 激活日志 - P:\PVDLOGS\PvDActivation.log.txt

最近的内容显示在日志文件结尾。

错误消息：7.6 及更高版本

以下错误消息对 PvD 7.6 及更高版本有效：

- **出现内部错误。 请查看 Personal vDisk 日志，进一步了解详细信息。 错误代码 %d (%s)**
此错误是对未分类错误的概括，因此没有数字值。在清单创建或 Personal vDisk 更新过程中遇到的所有异常错误均通过此错误代码指出。
 - 请收集日志并联系 Citrix 技术支持。
 - 如果此错误出现在目录更新过程中，请将目录回滚到之前的黄金映像版本。
- **规则文件中存在语法错误。 请查看日志，进一步了解详细信息。**
错误代码 2。规则文件包含语法错误。Personal vDisk 日志文件包含规则文件的名称和发现语法错误的行号。请修复规则文件中的语法错误，然后重试操作。
- **个人虚拟磁盘中存储的与早期版本的主映像对应的清单已损坏或无法访问。**
错误代码 3。最近的清单存储在 \ProgramData\CitrixPvD\Settings\Inventory\VER-LAST 下的 UserData.V2.vhd 中。请通过与早期主映像版本关联的已知可用的 PvD 计算机导入“VER-LAST”文件夹，还原与主映像的最近版本对应的清单。
- **个人虚拟磁盘中存储的与早期版本的主映像对应的清单版本较高。**
错误代码 4。此错误是由于个人虚拟磁盘版本在最近的主映像与当前主映像之间不兼容所致。请在主映像中安装最新个人虚拟磁盘版本后重新尝试更新目录。
- **Change journal overflow was detected. (检测到更改日志溢出。)**
错误代码 5。USN 日志溢出是由于在创建清单过程中对主映像进行了大量更改所致。如果在多次尝试后仍出现此情况，请使用 procmon 来确定是否存在第三方软件在清单创建过程中创建/删除大量文件的情况。
- **Personal vDisk 找不到连接到系统的用于存储用户数据的磁盘。**
错误代码 6。首先，通过虚拟机管理控制台验证 PvD 磁盘是否连接到 VM。此错误通常是由于“数据泄漏预防”软件阻止访问 PvD 磁盘所致。如果 PvD 磁盘已连接到 VM，请尝试在“数据泄漏预防”软件配置中添加“已连接磁盘”例外。
- **系统在安装后尚未重新启动。 请重新启动以使更改生效。**
错误代码 7。请重新启动桌面并重新尝试操作。
- **安装已损坏。 请尝试重新安装 Personal vDisk。**
错误代码 8。请安装 Personal vDisk 并重试。

- **个人虚拟磁盘清单不是最新的。请更新主映像中的清单，然后重试。**
 错误代码 9。关闭桌面前，个人虚拟磁盘清单在主映像中未更新。请重新启动主映像，并通过“更新 Personal vDisk”选项关闭桌面，然后创建新快照，并使用此快照更新目录。
- **启动个人虚拟磁盘时遇到内部错误。请查看 Personal vDisk 日志，进一步了解详细信息。**
 错误代码 10。出现此问题是由于内部错误或个人虚拟磁盘损坏导致 PVD 驱动程序无法启动虚拟会话所致。请尝试通过 Controller 重新启动桌面。如果问题仍然存在，请收集日志并联系 Citrix 技术支持。
- **尝试查找用于实现用户个性化设置的存储磁盘时 Personal vDisk 超时。**
 错误代码 11。如果 PVD 驱动程序在启动后 30 秒内未找到 PVD 磁盘，将会出现此错误。此问题通常是由于不受支持的 SCSI 控制器类型或存储冗余所致。如果目录中的所有桌面均出现此问题，请将“模板 VM”/“主 VM”关联的 SCSI 控制器类型更改为 Personal vDisk 技术支持的类型。如果此问题仅出现在目录中的部分桌面上，则可能是因为大量桌面在同一时间启动导致存储冗余出现峰值所致。请尝试限制与主机连接关联的活动电源操作设置最大值。
- **Personal vDisk 已取消激活，因为检测到不安全的系统关闭。请重新启动计算机。**
 错误代码 12。这可能是由于在启用 PVD 的情况下，桌面无法完成启动过程所致。请尝试重新启动桌面。如果问题仍然存在，请通过虚拟机管理控制台观察桌面启动情况，并检查桌面是否崩溃。如果桌面在启动过程中崩溃，请从备份还原 PVD（如有存在维护的备份）或重置 PVD。
- **为装载个人虚拟磁盘而指定的驱动器盘符不可用。**
 错误代码 13。出现此问题是因为在管理员指定的时间 PVD 无法装载 PVD 磁盘。如果驱动器盘符已由其他硬件使用，PVD 磁盘将无法装载。请选择其他盘符作为个人虚拟磁盘的装载点。
- **无法安装个人虚拟磁盘内核模式驱动程序。**
 错误代码 14。Personal vDisk 在安装后首次更新清单时安装驱动程序。如果从安装上下文以外尝试安装，有些防病毒产品会阻止驱动程序的安装。请在首次创建清单期间，临时禁止防病毒实时扫描，或在防病毒产品中将 PVD 驱动程序添加为例外。
- **无法创建系统卷的快照。请务必启用卷影复制服务。**
 错误代码 15。出现此错误可能是因为禁用了卷影复制服务。请启用卷影复制服务并尝试重新创建清单。
- **The change journal failed to activate. Try again after waiting for few minutes. (无法激活更改日志。请一段时间后重试。)**
 错误代码 16。Personal vDisk 使用更改日志来跟踪对主映像所做的更改。在清单更新期间，如果 PVD 检测到更改日志被禁用，PVD 会尝试将其启用；此尝试失败时会出现此错误。请稍等一段时间，然后重试。
- **系统卷上没有足够的可用空间。**
 错误代码 17。桌面的 C 驱动器上没有足够的可用空间来进行映像更新操作。请扩展系统卷，或删除系统卷中不再使用的文件以释放空间。映像更新应该会在下一次重新启动时重新开始。
- **个人虚拟磁盘存储上的可用空间不足。请扩展个人虚拟磁盘存储以提供更多空间。**
 错误代码 18。执行映像更新操作时，个人虚拟磁盘驱动器上的可用空间不足。请扩展个人虚拟磁盘存储或删除个人虚拟磁盘存储中不再使用的文件以释放空间。映像更新操作应该会在下次重新启动时重新开始。
- **个人虚拟磁盘存储已超负荷。请扩展个人虚拟磁盘存储以提供更多空间。**
 错误代码 19。个人虚拟磁盘驱动器上的可用空间不足，无法完全适应密集置备的“UserData.V2.vhd”。请扩展个人虚拟磁盘存储或删除个人虚拟磁盘存储中不再使用的文件以释放空间。
- **系统注册表已损坏。**
 错误代码 20。系统注册表损坏、缺失或不可读。重置个人虚拟磁盘或通过之前的备份还原。

- 重置个人虚拟磁盘时遇到内部错误。请查看 **Personal vDisk** 日志，进一步了解详细信息。
错误代码 21。这是对在个人虚拟磁盘重置期间遇到的所有错误的概括。请收集日志并联系 Citrix 技术支持。
- **Failed to reset the Personal vDisk because there is not enough free space in the personal vDisk storage.** (由于个人虚拟磁盘存储上的可用空间不足，无法重置个人虚拟磁盘。)
错误代码 22。执行重置操作时，个人虚拟磁盘驱动器上的可用空间不足。请扩展个人虚拟磁盘存储或删除个人虚拟磁盘存储中不再使用的文件以释放空间。

错误消息：7.6 之前的版本

以下错误消息对 7.6 之前的 PvD 7.x 版本有效：

- **启动失败。Personal vDisk 找不到用于存储用户个性化设置的存储磁盘。**
PvD 软件找不到个人虚拟磁盘（默认为驱动器 P:）或无法装载个人虚拟磁盘作为管理员在创建目录时选择的装载点。
 - 检查 PvD 服务日志中是否存在以下条目：PvD 1 status --> 18:183 (PvD 1 状态 --> 18:183)。
 - 如果您使用的是 5.6.12 之前的 PvD 版本，则升级到最新版本可解决此问题。
 - 如果您使用的是 5.6.12 版或更高版本，请使用磁盘管理工具 (diskmgmt.msc) 确定驱动器 P: 是否作为不可装入卷存在。如果存在，请在该卷上运行 chkdsk 以确定其是否已损坏，然后尝试使用 chkdsk 对其进行恢复。
- **启动失败。Citrix Personal vDisk 无法启动，如需更多帮助 状态代码：7，错误代码：0x70**
状态代码 7 表示尝试更新 PvD 时遇到错误。错误代码可以是以下代码之一：

错误代码：	说明
0x20000001	无法保存差异软件包，最可能的原因是 VHD 中可用磁盘空间不足。
0x20000004	无法获取更新 PvD 所需的权限。
0x20000006	无法从 PvD 映像或 PvD 清单加载配置单元，最可能的原因是 PvD 映像或清单已损坏。
0x20000007	无法加载文件系统清单，最可能的原因是 PvD 映像或清单已损坏。
0x20000009	无法打开包含文件系统清单的文件，最可能的原因是 PvD 映像或清单已损坏。
0x2000000B	无法保存差异软件包，最可能的原因是 VHD 中可用磁盘空间不足。
0x20000010	无法加载差异软件包。
0x20000011	规则文件丢失。
0x20000021	PvD 清单已损坏。
0x20000027	目录 MojoControl.dat 已损坏。
0x2000002B	PvD 清单已损坏或丢失。

0x2000002F 错误代码：	无法在映像更新时注册用户安装的 MOF，请升级到 5.6.12 以修复此问题。 说明
0x20000032	检查 PvDactivation.log.txt 中是否存在错误代码为 Win32 的最后一个日志条目。
0x20	无法装载应用程序容器以进行映像更新，请升级到 5.6.12 以修复此问题。
0x70	磁盘上空间不足。

- **启动失败。Citrix Personal vDisk 无法启动 [或 Personal vDisk 遇到内部错误]。如需更多帮助 ... 状态代码: 20，错误代码 0x20000028**

找到了个人虚拟磁盘，但无法创建 PvD 会话。

请收集日志并在 SysVol-IvmSupervisor.log 中查找是否存在会话创建失败：

1. 检查是否存在以下日志条目：IvmpNativeSessionCreate: failed to create native session, status XXXXX (IvmpNativeSessionCreate: 无法创建本机会话，状态为 XXXXX)。
2. 如果状态为 0xc00002cf，请通过将新版本的主映像添加到目录来修复此问题。此状态代码表示由于在更新清单后执行了大量更改，导致 USN 日志溢出。
3. 重新启动受影响的虚拟桌面。如果问题仍然存在，请联系 Citrix 技术支持。

- **启动失败。Citrix Personal vDisk 已取消激活，因为检测到不安全的系统关闭。要重试，请选择“重试”。如果问题继续存在，请联系系统管理员。**

在启用 PvD 的情况下，池 VM 无法完成其启动过程。请首先确定启动无法完成的原因。可能是因为由于以下原因显示蓝屏：

- 主映像中存在不兼容的防病毒产品，例如旧版 Trend Micro。
- 用户安装了与 PvD 不兼容的软件。这种可能性不大，但是您可以通过向目录中添加一个新计算机并观察它能否成功重新启动来进行检查。
- PvD 映像已损坏。在版本 5.6.5 中观察到此问题。

要检查池 VM 是否显示蓝屏或过早重新启动，请执行以下操作：

- 通过虚拟机管理程序控制台登录计算机。
- 单击重试并等待计算机关闭。
- 通过 Studio 启动计算机。
- 使用虚拟机管理程序控制台在计算机启动过程中对其进行监视。

其他故障排除方法：

- 从显示蓝屏的计算机中收集内存转储，然后将其发送给 Citrix 技术支持进行进一步分析。
- 检查事件日志中是否存在与 PvD 关联的错误：
 1. 使用 DiskMgmt.msc 通过单击 Action (操作) > Attach VHD (附加 VHD) 从驱动器 P: 的根目录中装载 UserData.V2.vhd。
 2. 启动 Eventvwr.msc。
 3. 通过单击 Action (操作) > Open saved logs (打开保存的日志) 从 UserData.V2.vhd 中打开系统事件日志 (Windows\System32\winevt\logs\system.evtx)。
 4. 通过单击 Action (操作) > Open saved logs (打开保存的日志) 从 UserData.V2.vhd 中打开应用程序事件日志 (Windows\System32\winevt\logs\application.evtx)。

- **Personal vDisk 无法启动。由于清单未更新，Personal vDisk 无法启动。请更新主映像中的清单，然后重试。状态代码：15，错误代码：0x0**

管理员在创建或更新 PvD 目录时选择了错误的快照（即，在创建快照时未使用更新 Personal vDisk 关闭主映像）。

Personal vDisk 记录的事件

如果未启用 Personal vDisk，可在 Windows 事件查看器中查看以下事件。请在左侧窗格中选择应用程序节点，右侧窗格中的事件来源为 Citrix Personal vDisk。如果启用 Personal vDisk，将不显示任何事件。

事件 ID 1 代表信息性消息，ID 2 代表错误。并非所有事件都能在所有版本的 Personal vDisk 中使用。

事件 ID	说明
1	Personal vDisk 状态：已启动清单更新。
1	Personal vDisk 状态：已完成清单更新。GUID：%s。
1	Personal vDisk 状态：已启动映像更新。
1	Personal vDisk 状态：已完成映像更新。
1	正在重置。
1	OK（正常）。
2	Personal vDisk 状态：清单更新失败，出现错误：%s。
2	Personal vDisk 状态：映像更新失败，出现错误：%s。
2	Personal vDisk 状态：映像更新失败，出现内部错误。
2	Personal vDisk 状态：清单更新失败，出现内部错误。
2	Personal vDisk 因不正常关机而被禁用。
2	映像更新失败。错误代码 %d。
2	Personal vDisk 出现内部错误。状态代码[%d] 错误代码[0x%X]。
2	Personal vDisk 重置失败。
2	找不到用于存储用户个性化设置的磁盘。
2	存储磁盘空间不足，无法创建 Personal vDisk 容器。

用户配置文件

May 04, 2016

默认情况下，在安装 Virtual Delivery Agent 时，Citrix Profile Management 以静默方式安装在主映像上，但您不必将 Profile Management 用作配置文件解决方案。

为迎合用户需求的不断变化，可使用 XenApp 和 XenDesktop 策略为每个交付组中的计算机应用不同的配置文件行为。例如，一个交付组可能需要 Citrix 强制配置文件（其模板保存在一个网络位置），而另一个交付组可能需要 Citrix 漫游配置文件（保存在包括多个重定向文件夹的其他位置）。

- 如果组织中的其他管理员负责 XenApp 和 XenDesktop 策略，应与他们合作以确保他们跨交付组设置任何配置文件相关的策略。
- 还可在组策略、Profile Management .ini 文件和各个本地虚拟机中设置 Profile Management 策略。定义配置文件行为的这些方式按照以下顺序读取：
 1. 组策略（.adm 或 .admx 文件）
 2. 策略节点中的 XenApp 和 XenDesktop 策略
 3. 用户连接的虚拟机上的本地策略
 4. Profile management .ini 文件

例如，如果您在组策略和策略节点中配置相同的策略，系统会读取组策略中的策略设置，而忽略 XenApp 和 XenDesktop 策略设置。

无论选择哪个配置文件解决方案，Director 管理员都可以访问这些用户配置文件的诊断信息并进行故障排除。有关详细信息，请参阅 Director 文档。

如果您使用 Personal vDisk 功能，Citrix 用户配置文件将默认存储在虚拟桌面的 Personal vDisk 中。不要在 Personal vDisk 上仍有配置文件副本时删除用户存储中的配置文件副本。这样做会造成 Profile Management 错误，并导致登录虚拟桌面将使用临时配置文件。

自动配置

会根据 Virtual Delivery Agent 安装自动检测桌面类型，并且，除了您在 Studio 中所做的配置选择，还会相应地设置 Profile Management 默认值。

Profile Management 调整的策略如下表所示。此功能将保留任何非默认策略设置，且不会将其覆盖。有关各策略的详细信息，请参阅 Profile Management 文档。创建配置文件的计算机类型会影响调整的策略。主要因素为计算机属于静态计算机还是预先置备的计算机，以及这些计算机是由多个用户共享还是专门仅供一个用户使用。

静态系统具有某些类型的本地存储，这些本地存储中的内容在关闭系统时有可能继续存在。静态系统可能会采用存储区域网络 (SAN) 等存储技术提供本地磁盘模仿。与此相反，预先置备的系统是基于基础磁盘和某些类型的身份磁盘即时创建的。本地存储通常通过 RAM 磁盘或网络磁盘进行模拟，网络磁盘通常由具有高速链路的 SAN 提供。置备技术通常为 Provisioning Services 或 Machine Creation Services（或第三方的等效技术）。预先置备的系统有时具有静态本地存储（可能由 Personal vDisk 提供）；此类计算机归类为静态计算机。

总而言之，这两类因素定义了以下计算机类型：

- **静态专用计算机** — 例如，具有静态分配以及通过 Machine Creation Services 创建的个人虚拟磁盘的桌面操作系统计算机、具有通过 VDI-in-a-Box 创建的个人虚拟磁盘的桌面、物理工作站和便携式计算机
- **静态共享计算机** — 例如，通过 Machine Creation Services 创建的服务器操作系统计算机
- **预先置备的专用计算机** — 例如，具有静态分配但不具有通过 Provisioning Services 创建的个人虚拟磁盘的桌面操作系统计算机

- **预先置备的共享计算机** — 例如，具有通过 Provisioning Services 创建的随机分配的桌面操作系统计算机、不具有通过 VDI-in-a-Box 创建的个人虚拟磁盘的桌面

以下 Profile Management 策略设置是针对不同的计算机类型建议的指导方针。这些设置在大多数情况下能够正常发挥作用，但根据部署要求，您可能希望使用与之有所差别的设置。

重要：注销时删除本地缓存的配置文件、Profile Streaming 和总是缓存是自动配置功能强制使用的策略。手动调整其他策略。
静态计算机

策略	静态专用计算机	静态共享计算机
注销时删除本地缓存的配置文件	已禁用	已启用
Profile Streaming	已禁用	已启用
始终缓存	已启用（备注 1）	已禁用（备注 2）
活动写回	已禁用	已禁用（备注 3）
处理本地管理员登录	已启用	已禁用（备注 4）

已置备的计算机

策略	预先置备的专用计算机	预先置备的共享计算机
注销时删除本地缓存的配置文件	已禁用（备注 5）	已启用
Profile Streaming	已启用	已启用
始终缓存	已禁用（备注 6）	已禁用
活动写回	已启用	已启用
处理本地管理员登录	已启用	已启用（备注 7）

1. 由于 Profile Streaming 对此类计算机禁用，因此将始终忽略总是缓存设置。
2. 禁用总是缓存。但是，可以通过启用此策略并用其定义一个文件大小限制 (MB) 来确保在登录后立即将大型文件加载到配置文件中。等于或大于此大小的任何文件都会立即在本地缓存。
3. 禁用活动写回，但在 XenApp 服务器间漫游的用户的配置文件中保存更改时除外。在这种情况下，请启用此策略。
4. 对除托管共享桌面以外的桌面禁用处理本地管理员登录。在这种情况下，请启用此策略。
5. 禁用注销时删除本地缓存的配置文件。这样将保留本地缓存的配置文件。由于计算机会在注销时重置，但分配给单个用户，

因此，如果缓存了其配置文件，登录速度将会更快。

6. 禁用总是缓存。但是，可以通过启用此策略并用其定义一个文件大小限制 (MB) 来确保在登录后立即将大型文件加载到配置文件中。等于或大于此大小的任何文件都会立即在本地缓存。
7. 启用处理本地管理员登录，但不对在 XenApp 和 XenDesktop 服务器间漫游的用户的配置文件启用此策略。在这种情况下，请禁用此策略。

文件夹重定向

通过文件夹重定向，可以将用户数据存储在配置文件的存储位置以外的网络共享上。这将减少配置文件大小和加载时间，但是可能会影响网络带宽。文件夹重定向不需要使用 Citrix 用户配置文件。您可以选择自己管理用户配置文件，仍可以重定向文件夹。

在 Studio 中使用 Citrix 策略配置文件夹重定向。

- 确保用于存储重定向文件夹内容的网络位置可用，并且具有适当的权限。已验证位置属性。
- 重定向文件夹已在网络上设置，并且已在登录时从用户的虚拟桌面填充其内容。

注意：仅使用 Citrix 策略或 Active Directory 组策略对象配置文件夹重定向，请勿同时使用二者。同时使用这两个策略引擎配置文件夹重定向可能会导致意外行为。

高级文件夹重定向

在包含多个操作系统的部署中，您可能希望每个操作系统共享用户的某些配置文件。其余配置文件则不共享，仅由一个操作系统使用。要确保在各个操作系统间提供一致的用户体验，需要对每个操作系统进行不同的配置。这就是高级文件夹重定向。例如，在两个操作系统上运行的应用程序的不同版本可能需要读取或编辑一个共享文件，因此您决定将共享文件重定向到两个版本均可访问的一个网络位置。或者，由于两个操作系统中的“开始”菜单文件夹内容在结构上有所不同，因此您决定仅重定向一个文件夹，而非两个文件夹。这将分隔每个操作系统上的“开始”菜单文件夹及其内容，从而确保一致的用户体验。

如果您的部署需要高级文件夹重定向，则必须了解用户配置文件数据的结构，并确定配置文件的哪些部分可在操作系统间共享。这一点非常重要，因为如果不正确使用文件夹重定向，可能会导致不可预测的行为。

在高级部署中重定向文件夹：

- 为每个操作系统使用单独的交付组。
- 了解虚拟应用程序（包括虚拟桌面上的虚拟应用程序），存储用户数据和设置的位置，并了解数据的结构。
- 对于可安全漫游（由于其结构在每个操作系统中均相同）的共享配置文件数据，请重定向每个交付组中的包含文件夹。
- 对于无法漫游的非共享配置文件数据，请仅重定向一个桌面组中的包含文件夹，通常选择使用最常用操作系统或其中的数据最相关的桌面组；对于无法在操作系统间漫游的非共享数据，请重定向两个系统中的包含文件夹以分隔网络位置。

高级部署示例 - 此部署中的一些应用程序（包括 Microsoft Outlook 和 Internet Explorer 版本）运行在 Windows 8 桌面上，另一些应用程序（包括 Outlook 和 Internet Explorer 的其他版本）则由 Windows Server 2008 交付。为实现此目的，您已为两个操作系统设置两个交付组。用户希望在这两个应用程序的两个版本中访问同一组联系人和收藏夹。

重要：以下决策和建议对介绍的操作系统和部署有效。在您的组织中，您选择重定向的文件夹以及是否决定共享这些文件夹取决于特定于您的具体部署的多种因素。

- 使用应用到交付组的策略选择下列要重定向的文件夹。

文件夹	已在 Windows 8 中重定向？	已在 Windows Server 2008 中重定向？
My Documents	是	是
Application Data	否	否
联系人	是	是

文件夹 桌面	已在 Windows 8 中重定向？ 是	已在 Windows Server 2008 中重定向？ 否
下载	否	否
收藏夹	是	是
链接	是	否
我的音乐	是	是
图片收藏	是	是
我的视频	是	是
搜索	是	否
保存的游戏	否	否
“开始”菜单	是	否

- 对于共享的重定向文件夹：
 - 在分析不同版本的 Outlook 和 Internet Explorer 保存的数据结构后，您确定可以安全共享“联系人”和“收藏夹”文件夹
 - 您知道“我的文档”、“我的音乐”、“图片收藏”和“我的视频”的结构在各个操作系统间均一致，因此将其存储在每个交付组的同一网络位置中非常安全
- 对于非共享的重定向文件夹：
 - 不重定向 Windows Server 交付组中的桌面、链接、搜索或“开始”菜单文件夹，因为这些文件夹中数据的组织结构在两个操作系统中有所不同，因此无法共享。
 - 要确保此非共享数据的可预测行为，只能在 Windows 8 交付组中重定向此数据。选择 Windows 8 而非 Windows Server 交付组是因为 Windows 8 在用户的日常工作中使用更频繁；他们只会偶尔访问服务器交付的应用程序。另外，在此情况下，非共享数据与桌面环境而非应用程序环境更为相关。例如，桌面快捷方式存储在“桌面”文件夹中，如果它们源自 Windows 8 计算机而非 Windows Server 计算机，则可能非常有用。
- 对于非重定向文件夹：
 - 您不希望使存储用户下载文件的服务器变得混乱，因此您选择不重定向“下载”文件夹
 - 来自单独应用程序的数据可能导致兼容性和性能问题，因此您决定不重定向“应用程序数据”文件夹

有关文件夹重定向的详细信息，请参阅 <http://technet.microsoft.com/en-us/library/cc766489%28v=ws.10%29.aspx>。

文件夹重定向和排除

在 Citrix Profile Management（而非 Studio）中，一项性能增强功能可防止使用排除处理文件夹。如果使用此功能，请不要排除任何重定向文件夹。文件夹重定向和排除功能配合使用，因此确保未排除重定向文件夹，可在您稍后决定不重定向这些文件夹时，让 Profile Management 再次将其移回配置文件文件夹结构，同时保持数据完整性。有关排除的详细信息，请参阅[包](#)

含和排除项目。

HDX

Nov 02, 2016

Citrix HDX includes a broad set of technologies that provide a high-definition user experience.

At the device	HDX leverages the computing capacity of user devices to enhance and optimize the user experience. HDX MediaStream technology ensures users receive a smooth, seamless experience with multimedia content in their virtual desktops or applications. Workspace control enables users to pause virtual desktops and applications and resume working from a different device at the point where they left off.
On the network	HDX incorporates advanced optimization and acceleration capabilities to deliver the best performance over any network, including low-bandwidth and high-latency WAN connections. HDX features adapt to changes in the environment, balancing performance and bandwidth by applying the best technologies for each unique user scenario, whether the desktop or application is accessed locally on the corporate network or remotely from outside the corporate firewall.
In the datacenter	HDX leverages the processing power and scalability of servers to deliver advanced graphical performance, regardless of the capabilities of the client device. Compressed multimedia information is sent directly to the user device in its native format. HDX channel monitoring provided by Citrix Director displays the status of connected HDX channels on user devices. HDX Insight, the integration of EdgeSight Network Inspector and EdgeSight Performance management with Director, captures data about ICA traffic and provides a dashboard view of real-time and historical details such as client-side and server-side ICA session latency, bandwidth use of ICA channels, and the ICA round trip time value of each session.

To experience HDX capabilities from your virtual desktop:

- See how HDX delivers rich video content to virtual desktops: View a video on a web site containing high definition videos, such as <http://www.microsoft.com/silverlight/iis-smooth-streaming/demo/>.
- See how Flash Redirection accelerates delivery of Flash multimedia content:
 1. Download Adobe Flash player (<http://get.adobe.com/flashplayer/>) and install it on both the virtual desktop and the user device.
 2. On the Desktop Viewer toolbar, click Preferences. In the Desktop Viewer Preferences dialog box, click the Flash tab and select Optimize content.
 3. To experience how Flash Redirection accelerates the delivery of Flash multimedia content to virtual desktops, view a video on your desktop from a web site containing Flash videos, such as YouTube. Flash Redirection is designed to be seamless so that users do not know when it is running. You can check to see whether Flash Redirection is being used by looking for a block of color that appears momentarily before the Flash player starts.
- See how HDX delivers high definition audio:
 1. Configure your Citrix client for maximum audio quality; see the Receiver documentation for details.
 2. Play music files with a digital audio player (such as iTunes) on your desktop.

HDX provides a superior graphics and video experience for most users by default, with no configuration required. Citrix

policy settings that provide the best out-of-the-box experience for the majority of use cases are enabled by default.

- HDX automatically selects the best delivery method based on the client, platform, application, and network bandwidth, and then self-tunes based on changing conditions.
- HDX optimizes the performance of 2D and 3D graphics and video.
- HDX delivers a Windows Aero experience to virtual desktop users on any client.
- HDX enables user devices to stream multimedia files directly from the source provider on the Internet or Intranet, rather than through the host server. If the requirements for this client-side content fetching are not met, media delivery falls back to Windows Media redirection to play media run-time files on user devices rather than the host server. In most cases, no adjustments to the Windows Media feature policies are needed.

Good to know:

- For support and requirements information for HDX features, see [System requirements for XenApp and XenDesktop 7.6](#). Except where otherwise noted, HDX features are available for supported Windows Server OS and Windows Desktop OS machines, plus Remote PC Access desktops.
- This content describes how to further optimize the user experience, improve server scalability, or reduce bandwidth requirements. For information about working with Citrix policies and policy settings, see the *— Citrix policies* documents for this release.
- For instructions that include working with the registry, use caution: editing the registry incorrectly can cause serious problems that may require you to reinstall your operating system. Citrix cannot guarantee that problems resulting from the incorrect use of Registry Editor can be solved. Use Registry Editor at your own risk. Be sure to back up the registry before you edit it.

Reduce the bandwidth needed for Windows desktops

By default, HDX delivers a highly responsive Windows Aero or Windows 8 desktop experience to virtual desktops accessed from supported Windows user devices. To do that, HDX leverages the graphics processing unit (GPU) or integrated graphics processor (IGP) on the user devices for local DirectX graphics rendering. This feature, named desktop composition redirection, maintains high scalability on the server. For details, see What to do with all these choices in <http://blogs.citrix.com/2013/11/06/go-supersonic-with-xendesktop-7-x-bandwidth-supercodecs/>.

To reduce the bandwidth required in user sessions, consider adjusting the following Citrix policy settings. Keep in mind that changing these settings can reduce the quality of the user experience.

- **Desktop Composition Redirection.** Applies only to Windows Desktop OS machines accessed from Windows user devices and applies only to the composition of the Windows desktop. Application windows are rendered on the server unless the Citrix policy setting Allow local app access is Allowed.
- **Desktop Composition Redirection graphics quality.** Uses high-quality graphics for desktop composition unless seamless applications or Local App Access are enabled. To reduce bandwidth requirements, lower the graphics quality.
- **Dynamic windows preview.** Controls the display of seamless windows in Flip, Flip 3D, taskbar preview, and peek window preview modes. To reduce bandwidth requirements, disable this policy setting.

Improve the image quality sent to user devices

The following visual display policy settings control the quality of images sent from virtual desktops to user devices.

- **Visual quality.** Controls the visual quality of images displayed on the user device: medium, high, always lossless, build to lossless (default = medium).
- **Target frame rate.** Specifies the maximum number of frames per second that are sent from the virtual desktop to the user device (default = 30). In many circumstances, you can improve the user experience by specifying a higher value. For devices with slower CPUs, specifying a lower value can improve the user experience.

- Display memory limit. Specifies the maximum video buffer size for the session in kilobytes (default = 65536 KB). For connections requiring more color depth and higher resolution, increase the limit. You can calculate the maximum memory required. Color depths other than 32-bit are available only if the Legacy graphics mode policy setting is enabled.

Improve video conference performance

HDX webcam video compression improves bandwidth efficiency and latency tolerance for webcams during video conferencing in a session. This technology streams webcam traffic over a dedicated multimedia virtual channel; this uses significantly less bandwidth compared to the isochronous HDX Plug-n-Play support, and works well over WAN connections.

Receiver users can override the default behavior by choosing the Desktop Viewer Mic & Webcam setting Don't use my microphone or webcam. To prevent users from switching from HDX webcam video compression, disable USB device redirection with the policy settings under ICA policy settings > USB Devices policy settings.

HDX webcam video compression is enabled by default on Receiver for Windows but must be configured on Receiver for Linux. For more information, refer to the Receiver documentation. HDX webcam video compression requires that the following policy settings be enabled (all are enabled by default).

- Client audio redirection
- Client microphone redirection
- Multimedia conferencing
- Windows Media Redirection

If a webcam supports H.264 hardware encoding, HDX video compression uses the hardware encoding by default. Hardware encoding uses additional bandwidth and is not suitable for a low bandwidth network. To force software compression over low bandwidth networks, add the following DWORD key value to the registry key: HKCU\Software\Citrix\HdxRealTime: DeepCompress_ForceSWEncode=1.

Choose server scalability over user experience

For deployments where server scalability is of greater concern than user experience, you can use the legacy graphics system by adding the Legacy graphics mode policy setting and configuring the individual legacy graphics policy settings. Use of the legacy graphics system affects the user experience over WAN and mobile connections.

Thinwire 兼容模式

Nov 02, 2016

This article applies to 7.6 FP3.

Thinwire Compatibility Mode uses new screen decomposition and caching techniques, which achieve low bandwidth usage and high server scalability without compromising the end-user experience.

Thinwire Compatibility Mode includes the following features:

- Intelligent bitmap matching for a bitmap-only provider.
 - Bitmap translation analysis for efficient window movement and scrolling.
- Backwards compatible. There is no requirement for client or Citrix Receiver upgrades or hardware acceleration.
 - Tested on a range of older thin clients up to and over 5 years old.
- Optimized for very low server CPU usage and improved server scalability.
- An emulated 16-bit mode, which reduces bandwidth by a further 15-20% for typical workloads.
- Transient detection for server-rendered video content.
 - Multi-transient handling for an improved multimedia experience. For example, when watching multiple videos or ticker tapes.
 - Selective sharpening for regions that leave a transient state.
- Optimized for CloudBridge acceleration. In tests, we have seen up to a 6:1 ratio of bandwidth reduction on Office-type workloads.
- Adaptive display, which can be tuned through policy settings. For more information see **Moving image compression** in [Moving image policy settings](#).
- VDA's and Windows OS's up to and including Windows 10 VDA are supported.
- New "Build to Lossless" mode for 3D Pro, which improves responsiveness, interactivity, and interruptible sharpening for a better user experience on low bandwidth.
- Default static photographic imagery quality is higher than in Legacy Graphics Mode.

For Visual Quality settings "Low", "Medium" (default) and "High", the transient detector dynamically evaluates screen updates to decide whether highly-animated areas should be sent at lower quality, in accordance with the Adaptive Display policy, to improve client performance and reduce bandwidth usage.

For the **Build to lossless** visual quality, Thinwire Compatibility Mode uses a "fuzzy-first" approach for large screen updates. This setting is targeted at 3D Pro users who are manipulating 3D models or other graphic-intensive applications. If the activity continues, a transient mode is assumed and the affected area is sharpened and cached once transient activity stops. For the initial large change, some lightweight image analysis is performed on the change area to determine whether to use "fuzzy transient" or "sharp transient" (lossless) - for example, when rotating a wireframe. It is more efficient, for FPS (Frames Per Second) and bandwidth, to encode simple imagery using the Citrix lossless codec and no loss in quality occurs.

The sharpen-to-lossless step in Build to lossless is also different. Rather than sharpening the affected area in one step, the area is sharpened in pre-determined blocks to help maintain interactivity and a smooth user experience. Sharpening a large change area mid-transient, for example moving a 3D model which is stopped briefly, then moved again, would previously cause a "stall", especially over a low bandwidth line. The size of the sharpening blocks depends on how far the quality was reduced to try and maintain the target minimum frame rate, which is an Adaptive Display policy setting. If the quality was significantly reduced, the sharpening block size will be smaller, with a minimum size of 128 x 128 pixels. If the quality was not reduced, for example, when the client has adequate processing power and bandwidth, the sharpening block size can be a

maximum size of 384 x 384 pixels.

HDX 3D Pro

Nov 02, 2016

HDX 3D Pro enables you to deliver desktops and applications that perform best with a graphics processing unit (GPU) for hardware acceleration, including 3D professional graphics applications based on OpenGL and DirectX. (The standard VDA supports GPU acceleration of DirectX only.)

Examples of 3D professional applications include:

- Computer-aided design, manufacturing, and engineering (CAD/CAM/CAE) applications
- Geographical Information System (GIS) software
- Picture Archiving Communication System (PACS) for medical imaging
- Applications using the latest OpenGL, DirectX, NVidia CUDA, and OpenCL versions
- Computationally-intensive non-graphical applications that use NVIDIA Compute Unified Device Architecture (CUDA) GPUs for parallel computing

HDX 3D Pro provides the best user experience over any bandwidth:

- On wide area network (WAN) connections: Deliver an interactive user experience over WAN connections with bandwidths as low as 1.5 Mbps.
- On local area network (LAN) connections: Deliver a user experience equivalent to that of a local desktop on LAN connections with bandwidths of 100 Mbps.

You can replace complex and expensive workstations with simpler user devices by moving the graphics processing into the data center for centralized management.

HDX 3D Pro provides GPU acceleration for Windows Desktop OS machines and Windows Server OS machines. When used with Citrix XenServer and NVIDIA GRID GPUs, HDX 3D Pro provides Virtual GPU (vGPU) acceleration for Windows Desktop OS machines. For the supported XenServer versions, see [Citrix Virtual GPU Solution](#).

Use the HDX Monitor tool (which replaces the Health Check tool) to validate the operation and configuration of HDX visualization technologies and to diagnose and troubleshoot HDX issues. To download the tool and learn more about it, see <https://taas.citrix.com/hdx/download/>.

Flash 重定向

May 04, 2016

Flash 重定向功能可将大部分 Adobe Flash 内容（包括动画、视频和应用程序）处理工作转移到连接至 LAN 和 WAN 的用户 Windows 设备，从而降低服务器和网络的工作负载。这样将提供更高的可扩展性，同时确保获得高清晰度用户体验。配置 Flash 重定向时，必须同时进行服务器端设置和客户端设置。

警告：使用 Flash 重定向功能时会在用户设备和服务器组件之间进行大量交互。只应在用户设备和服务器之间无需安全分隔的环境中使用此功能。此外，用户设备应配置为只在可信服务器上使用此功能。由于 Flash 重定向功能要求在用户设备上安装 Flash Player，因此，只有在 Flash Player 自身处于安全状态时才应启用 Flash 重定向。

旧版和第二代 Flash 重定向功能是独立解决方案，并在单独的虚拟通道中运行。

- 旧版 Flash 重定向功能仅在客户端受支持。如果用户设备上安装了早期版本的 Flash Player，或者无法安装 Flash Player，Flash 内容将在服务器上呈现。
- 第二代 Flash 重定向功能在客户端和服务器上均受支持。如果客户端支持第二代 Flash 重定向功能，Flash 内容将在客户端上呈现。第二代 Flash 重定向功能包括支持用户通过 WAN 进行连接、智能回退和 URL 兼容性列表；请参阅下文了解详情。

Flash 重定向使用服务器上的 Windows 事件日志记录来记录 Flash 事件。事件日志将指示是否正在使用 Flash 重定向，并提供问题的详细信息。Flash 重定向记录的所有事件都具有以下共性：

- Flash 重定向向应用程序日志报告事件。
- 在 Windows 8 和 Windows 7 系统上，特定于 Flash 重定向功能的日志将显示在“应用程序和服务日志”节点中。
- “源”值为 Flash。
- “类别”值为“无”。

有关 HDX Flash 最新更新的兼容性，请参阅 [CTX136588](#)。

要在服务器上配置 Flash 重定向，请使用以下 Citrix 策略设置。有关详细信息，请参阅 [Flash 重定向策略设置](#)。

- Flash 默认行为可确定默认的 Flash 加速行为。默认情况下，Flash 重定向已启用。要为个别 Web 页面和 Flash 实例覆盖此默认行为，请使用 Flash URL 兼容性列表设置。
- Flash 智能回退 - 检测 Flash 小电影的实例（例如常用于播放广告的 Flash 小电影），并将这些实例呈现在服务器上，而不是通过重定向呈现在用户设备上。此功能不会在加载 Web 页面或 Flash 应用程序时导致任何中断或故障。默认情况下，启用 Flash 智能回退。要重定向所有 Flash 内容实例，使其呈现在用户设备上，请禁用此策略设置。
- Flash 服务器端内容提取 URL 列表允许您指定一些 Web 站点，这些站点的 Flash 内容可以下载到服务器，然后传输到用户设备以进行呈现。（默认情况下，Flash 重定向会将 Flash 内容下载到用于播放此内容的用户设备。）此设置与用户设备上的启用服务器端内容提取设置结合使用，主要用于 Intranet 站点和内部 Flash 应用程序；请参阅下文了解详情。此设置还可以用于大多数 Internet 站点，可以在用户设备不直接访问 Internet 时（例如，由 XenApp 或 XenDesktop 服务器提供连接时）使用。

注意：服务器端内容提取功能不支持使用实时消息传送协议 (RTMP) 的 Flash 应用程序；此时应改为使用服务器端呈现功能，该功能支持 HTTP 和 HTTPS。

- Flash URL 兼容性列表 - 指定在何处呈现来自所列 Web 站点的 Flash 内容：在用户设备上、服务器上还是被阻止。
- Flash 背景色列表 - 可以将 Web 页面与 Flash 实例的颜色相匹配，从而在使用 Flash 重定向时改进 Web 页面的外观。

在用户设备上安装 Citrix Receiver 和 Adobe Flash Player。无需在用户设备上执行任何其他配置。

可以使用 Active Directory 组策略对象更改默认设置。导入并添加 HDX MediaStream Flash 重定向 - 客户端管理模板 (HdxFlashClient.adm)，位于以下文件夹：

- 对于 32 位计算机：%Program Files%\Citrix\ICA Client\Configuration\语言

- 对于 64 位计算机：%Program Files (x86)%\Citrix\ICA Client\Configuration\语言

策略设置显示在管理模板 > 经典管理模板(ADM) > HDX MediaStream Flash 重定向- 客户端下。请参阅 Microsoft Active Directory 文档了解 GPO 和模板的详细信息。

更改 Flash 重定向功能的使用时间

在用户设备上启用 HDX MediaStream Flash 重定向策略设置与服务器端设置结合使用，可控制是否将 Adobe Flash 内容重定向到用户设备，以便在本地呈现。默认情况下，Flash 重定向已启用，并使用智能网络检测功能来确定何时在用户设备上播放 Flash 内容。

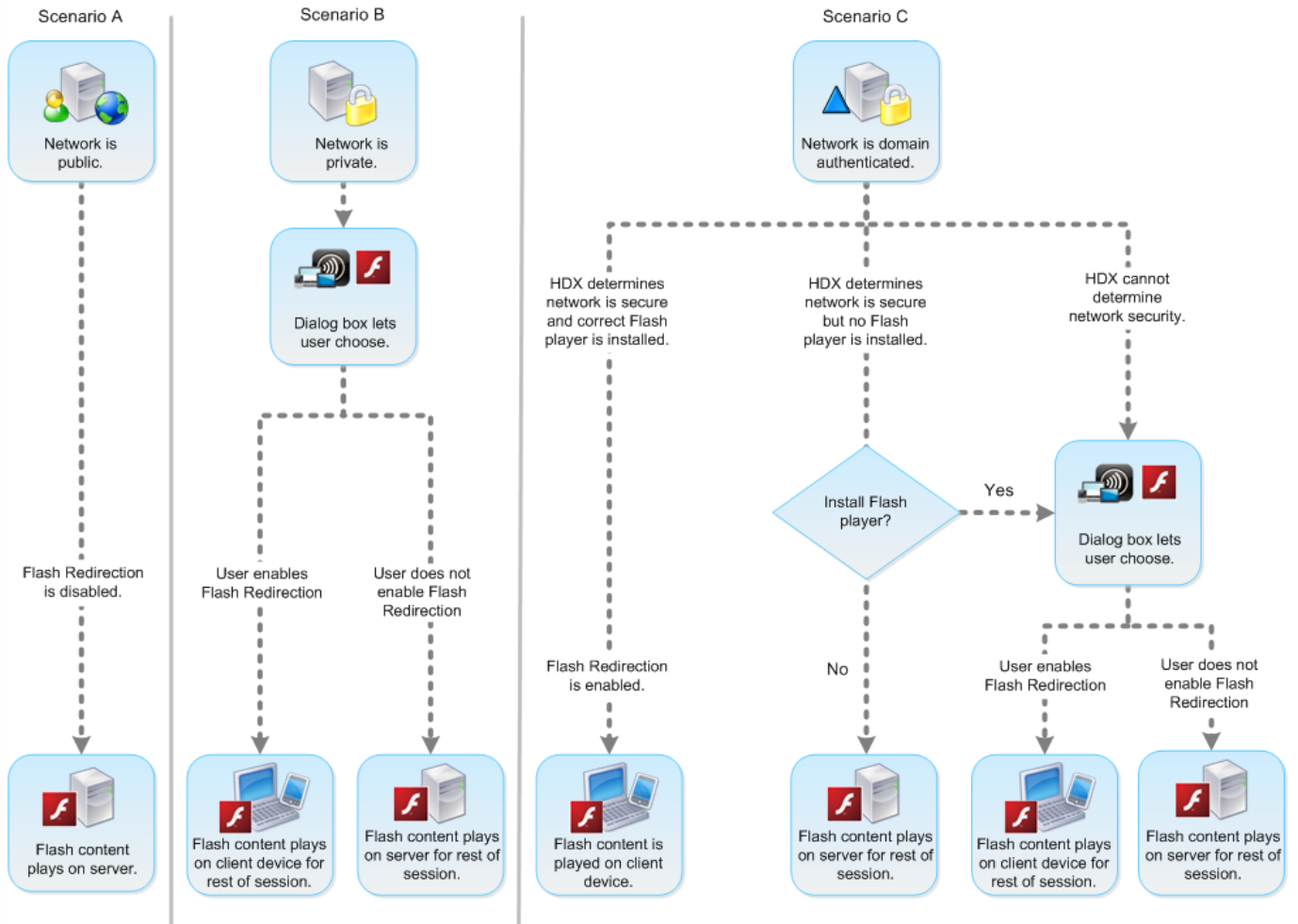
如果未设置配置并且使用 Desktop Lock，则默认情况下，将在用户设备上启用 Flash 重定向。

要更改 Flash 重定向功能的使用时间，或在用户设备上禁用 Flash 重定向功能：

1. 从“设置”列表中，选择在用户设备上启用 HDX MediaStream Flash 重定向，然后单击策略设置。
2. 选择未配置、启用（默认设置）或禁用。
3. 如果选择启用，请从使用 HDX MediaStream Flash 重定向列表中选择一个选项：
 - 要在具有所需配置时使用最新 Flash 重定向功能，并在没有此配置时还原为在服务器端呈现，请选择仅使用第二代选项。
 - 要始终使用 Flash 重定向，请选择始终。将在用户设备上播放 Flash 内容。
 - 要始终不使用 Flash 重定向，请选择从不。将在服务器上播放 Flash 内容。
 - 要使用智能网络检测功能评估客户端网络的安全级别，以确定何时使用 Flash 重定向功能，请选择询问（默认设置）。如果无法确定网络安全性，系统会询问用户是否使用 Flash 重定向。如果无法确定网络安全级别，系统会提示用户选择是否使用 Flash 重定向。

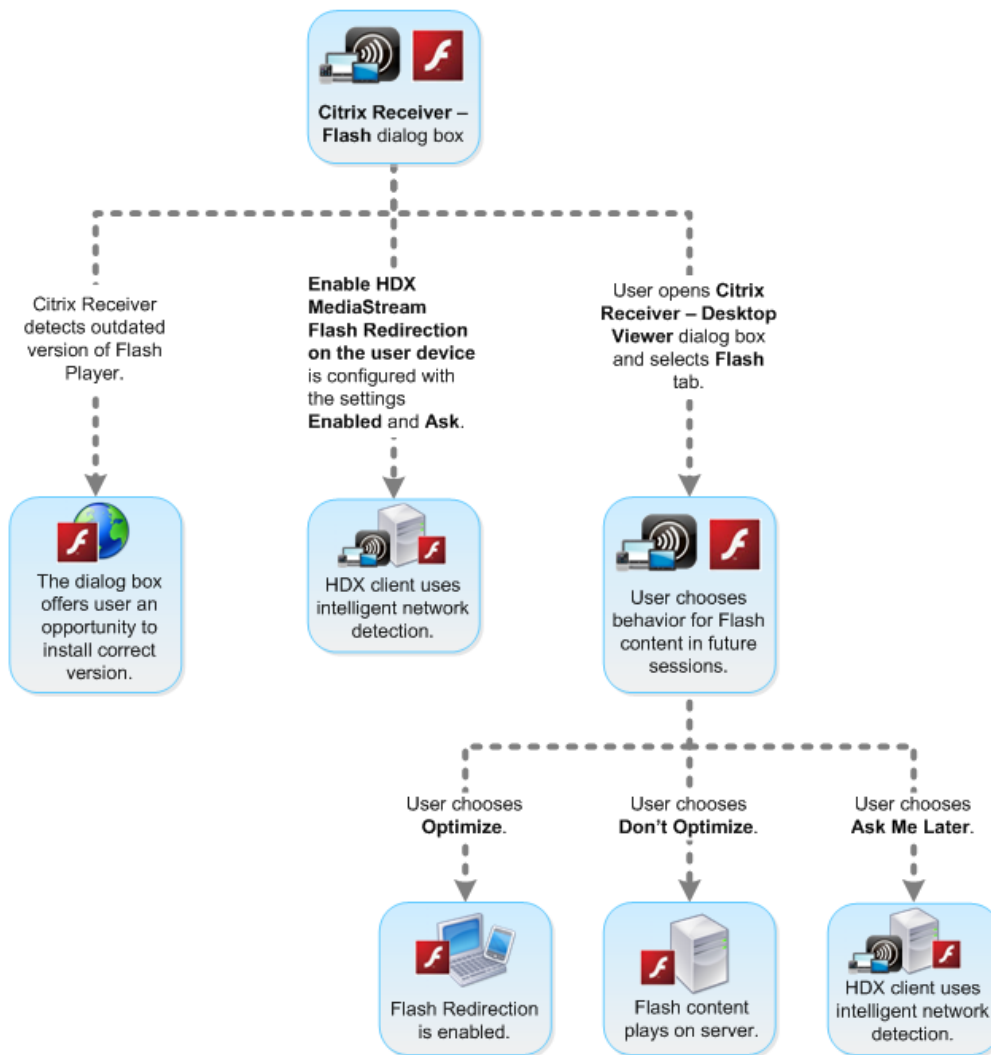
下图显示了 Flash 重定向针对各种网络类型的处理方式。

Intelligent Network Detection for Flash Redirection



用户可以从 Citrix Receiver - Desktop Viewer 首选项对话框的 Flash 选项卡中选择优化或不优化，来覆盖智能网络检测。根据用户设备上的 Flash 重定向配置，可选择的选项将有所不同，如下图所示。

User control of Flash redirection



在客户端与服务器端之间同步 HTTP Cookie

默认情况下，客户端与服务器端之间的 HTTP Cookie 同步已禁用。启用同步功能，以从服务器下载 HTTP Cookie；这些 HTTP Cookie 用于客户端内容提取，并可供包含 Flash 内容的站点在需要时使用。

注意：客户端 Cookie 在同步期间不会被替换；即使之后同步策略被禁用，它们也将保持可用。

1. 从“设置”列表中，选择启用客户端与服务器端之间的 HTTP Cookie 同步，然后单击策略设置。
2. 选择未配置、启用或禁用（默认设置）。

启用服务器端内容提取

默认情况下，Flash 重定向功能会将 Adobe Flash 内容下载到用于播放此内容的用户设备。启用服务器端内容提取会使 Flash 内容下载到服务器上，然后再发送到用户设备。除非有覆盖策略（例如通过 Flash URL 兼容性列表策略设置阻止某个站点），否则 Flash 内容将在用户设备上播放。

当用户设备通过 NetScaler Gateway 连接到内部站点以及用户设备没有直接访问 Internet 的权限时，会频繁使用服务器端内容提取。

注意：服务器端内容提取功能不支持使用实时消息传送协议 (Real Time Messaging Protocols, RTMP) 的 Flash 应用程序。而应对此类站点使用服务器端呈现功能。

第二代 Flash 重定向功能支持服务器端内容提取功能的三个启用选项。其中两个选项包含在用户设备上缓存服务器端内容的功能，这样一来，由于重复使用的内容已经存储在用户设备上用于呈现，因而提高了性能。该缓存的内容与用户设备上

缓存的其他 HTTP 内容存储在不同的位置。

使用第二代 Flash 重定向时，如果任一启用选项被选中并且客户端 .swf 文件提取操作失败，将自动开始执行服务器端内容提取回退操作。

要启用服务器端内容提取功能，必须同时在客户端设备和服务器上进行设置。

1. 从“设置”列表中，选择启用服务器端内容提取，然后单击策略设置。
2. 选择未配置、启用或禁用（默认设置）。如果启用此设置，请从服务器端内容提取状态列表选择一个选项：

选项	说明
已禁用	禁用服务器端内容提取，以覆盖服务器上的 Flash 服务器端内容提取 URL 列表设置。服务器端内容提取回退也已禁用。
已启用	为 Flash 服务器端内容提取 URL 列表中所标识的 Web 页面和 Flash 应用程序启用服务器端内容提取。可以使用服务器端内容提取回退功能，但是不会缓存 Flash 内容。
已启用 (永久 缓存)	为 Flash 服务器端内容提取 URL 列表中所标识的 Web 页面和 Flash 应用程序启用服务器端内容提取。可以使用服务器端内容提取回退。通过服务器端提取而获得的内容缓存在用户设备上，并在会话间存储。
启用 (临时 缓存)	为 Flash 服务器端内容提取 URL 列表中所标识的 Web 页面和 Flash 应用程序启用服务器端内容提取。可以使用服务器端内容提取回退。通过服务器端提取而获得的内容缓存在用户设备上，并在会话结束时删除。

3. 在服务器上，启用 Flash 服务器端内容提取 URL 列表策略设置，并在其中填写目标 URL。

将用户设备重定向到其他服务器以实现客户端内容提取

可以使用客户端内容提取 URL 重写规则设置（第二代 Flash 重定向功能）将获取 Flash 内容的尝试重定向。配置此功能时，需要提供两个 URL 模式；如果用户设备尝试从与第一种模式（URL 匹配模式）相匹配的 Web 站点提取内容，会被重定向到由第二种模式（重写的 URL 格式）所指定的 Web 站点。

可以使用此设置作为内容交付网络 (CDN) 的补充。一些交付 Flash 内容的 Web 站点使用 CDN 重定向，使用户能够从包含相同内容的最近一组服务器获得内容。使用 Flash 重定向客户端内容提取功能时，将从用户设备请求 Flash 内容，而 Flash 内容所在的其余 Web 页面则由服务器请求。如果在使用 CDN，服务器请求会重定向到最近的服务器，用户设备请求也会到达同一位置。这可能并不是距离用户设备最近的位置；在加载 Web 页面和播放 Flash 内容之间可能存在明显的延迟，具体取决于距离。

1. 从“设置”列表中，选择客户端内容提取的 URL 重写规则，然后单击策略设置。
2. 选择未配置、已启用或已禁用。未配置是默认设置；选择已禁用将忽略在下一步中配置的任何 URL 重写规则。
3. 如果启用设置，请单击显示。使用 Perl 正则表达式语法在值名称框中键入 URL 匹配模式，并在值框中键入重写的 URL 格式。

在 XenApp 和 XenDesktop 7.6 FP3 中，您可以添加注册表设置，以指定使用 Receiver for Windows 或 Receiver for Linux 访问 VDA 的客户端设备进行 Flash 重定向所需的最低版本。

警告

注册表编辑不当会导致严重问题，可能导致需要重新安装操作系统。Citrix 无法保证因“注册表编辑器”使用不当导致出现的问题能够得以

解决。使用“注册表编辑器”需自担风险。在编辑注册表之前，请务必进行备份。

ServerFlashPlayerVersionMinimum 是一个字符串值，指定 ICA 服务器 (VDA) 上所需 Flash Player 的最低版本。

ClientFlashPlayerVersionMinimum 是一个字符串值，指定 ICA 客户端 (Citrix Receiver) 上所需 Flash Player 的最低版本。

这些版本字符串可以采用“10”、“10.2”或“10.2.140”格式指定。目前，仅比较主版本号、次要版本号和内部版本号。修订版本号将被忽略。例如，指定“10”的版本字符串表示仅指定主版本号，次要版本号和内部版本号假定为零。

FlashPlayerVersionComparisonMask 是一个 DWORD 值，设置为零将禁止比较 ICA 客户端上的 Flash Player 与 ICA 服务器上的 Flash Player 的版本。比较掩码具有其他值，但由于任何非零掩码的意思可能会改变，因此不应使用。建议仅为所需的客户端将比较掩码设置为零。建议不要在客户端诊断设置下设置比较掩码。如果未指定比较掩码，Flash 重定向将要求 ICA 客户端上的 Flash Player 版本高于或等于 ICA 服务器上的 Flash Player 版本。此功能仅比较 Flash Player 的主要版本号来实现此要求。

要使重定向发生，除了使用比较掩码进行检查，还需要成功完成客户端和服务器的最低版本检查。

子项 ClientID0x51 指定 Linux ICA 客户端。子项 ClientID0x1 指定 Windows ICA 客户端。此子项的命名方式是在字符串“ClientID”后面附加十六进制客户端产品 ID（无任何前导零）。Mobile SDK for Windows Apps 文档 <http://www.citrix.com/mobilitysdk/docs/clientdetection.html> 中提供了客户端 ID 的完整列表

32 位 VDA 注册表配置示例

[HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\HdxMediaStreamForFlash\Server\PseudoServer] 客户端诊断设置

"ClientFlashPlayerVersionMinimum"="13.0" ICA 客户端所需的最低版本 "ServerFlashPlayerVersionMinimum"="13.0" ICA 服务器所需的最低版本 [HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\HdxMediaStreamForFlash\Server\PseudoServer\ClientID0x1] Windows ICA 客户端设置

"ClientFlashPlayerVersionMinimum"="16.0.0" 指定 Windows 客户端所需的 Flash Player 的最低版本

[HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\HdxMediaStreamForFlash\Server\PseudoServer\ClientID0x51] Linux ICA 客户端设置

"FlashPlayerVersionComparisonMask"=dword:00000000 为 linux 客户端禁用版本比较检查（检查以确定客户端上的 Flash Player 版本高于服务器上的版本） "ClientFlashPlayerVersionMinimum"="11.2.0" 此设置指定 Linux 客户端的 Flash Player 的最低版本。

64 位 VDA 注册表配置示例

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer]

"ClientFlashPlayerVersionMinimum"="13.0" "ServerFlashPlayerVersionMinimum"="13.0"

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer\ClientID0x1]

"ClientFlashPlayerVersionMinimum"="16.0.0"

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer\ClientID0x51]

"FlashPlayerVersionComparisonMask"=dword:00000000 "ClientFlashPlayerVersionMinimum"="11.2.0"

Host to client redirection

Nov 02, 2016

Content redirection allows you to control whether users access information with applications published on servers or with applications running locally on user devices.

Host to client redirection is one kind of content redirection. It is supported only on Server OS VDAs (not Desktop OS VDAs).

- When host to client redirection is enabled, URLs are intercepted at the server VDA and sent to the user device. The web browser or multimedia player on the user device opens these URLs.
- If you enable host to client redirection and the user device fails to connect to a URL, the URL is redirected back to the server VDA.
- When host to client redirection is disabled, users open the URLs with web browsers or multimedia players located on the server VDA.
- When host to client redirection is enabled, users cannot disable it.

Host to client redirection was previously known as **server to client redirection**.

You might consider using host to client redirection in specific but uncommon cases, for performance, compatibility, or compliance. Normally, other forms of content redirection are better.

Performance

You can use host to client redirection for performance, so that whenever an application is installed on the user device, it is used in preference to an application on the VDA.

Keep in mind that host to client redirection will improve performance only under specific conditions, because the VDA already optimizes Adobe Flash and other types of multimedia content. First, consider using the other approaches (policy settings) noted in the tables below, rather than host to client redirection; they offer more flexibility and usually give a better user experience, particularly for less-powerful user devices.

Compatibility

You can use host to client redirection for compatibility in the following use cases:

- You use content types other than HTML or multimedia (for example, a custom URL type).
- You use a legacy media format (such as Real Media) that is not supported by the VDA's multimedia player with multimedia redirection.
- The application for the content type is used by only a small number of users who already have the application installed on their user device.
- The VDA cannot access certain web sites (for example, web sites internal to another organization).

Compliance

You can use host to client redirection for compliance in the following use cases:

- The application or content licensing agreement does not permit publishing via the VDA.

- Organizational policy does not permit a document being uploaded to the VDA.

Some situations are more likely in complex environments, and also if the user device and the VDA belong to different organizations.

Environments may have many different types of user devices.

User device	Situation or environment	Content redirection approach
Tablet	-	Any approach (see next table)
Laptop PC	-	Any approach (see next table)
Desktop PC	Users use a wide range of apps installed on the user device	Any approach (see next table)
Desktop PC	Users use only a few known apps that are installed on the user device	Local App Access
Desktop PC	Users use no apps installed on the user device	Multimedia redirection and/or Flash redirection
Desktop appliance	Vendor supports multimedia redirection and/or Flash redirection	Multimedia redirection and/or Flash redirection
Thin client	Vendor supports multimedia redirection, Flash redirection, and host to client redirection	Any approach (see next table)
Zero client	Vendor supports multimedia redirection and/or Flash redirection	Multimedia redirection and/or Flash redirection

Use the following examples to help guide your content redirection approach.

URLs link	Situation or environment	Content redirection approach
A web page or document	The VDA cannot access the URL	Host to client redirection
A web page	The web page contains Adobe Flash	Flash redirection

A multimedia file or stream	The VDA has a compatible multimedia player	Multimedia redirection
A multimedia file or stream	The VDA does not have a compatible multimedia player	Host to client redirection
A document	The VDA does not have an application for that document type	Host to client redirection
A document	The document must not be downloaded to the user device	No redirection
A document	The document must not be uploaded to the VDA	Host to client redirection
A custom URL type	The VDA does not have an application for that custom URL type	Host to client redirection

Host to client redirection is supported by Citrix Receiver for Windows, Receiver for Mac, Receiver for Linux, Receiver for HTML5, and Receiver for Chrome.

To use host to client redirection, the user device must have a web browser, multimedia player, or other application that is suitable for the content. If the user device is a desktop appliance, thin client, or zero client, confirm that it has suitable applications and is sufficiently powerful.

User devices enabled for Local App Access use a different mechanism for content redirection, and do not require host to client content redirection.

You can use Citrix policies to prevent host to client content redirection for unsuitable devices.

Host to client redirection is used when URLs are:

- Embedded as hyperlinks in an application (for example, in an email message or document).
- Selected through a VDA application's menus or dialogs, provided that the application uses the Windows ShellExecuteEx API.
- Entered in the Windows Run dialog.

Host to client redirection is not used for URLs in a web browser (either in a web page or entered in the address bar of the web browser).

注意

If users change their default web browser on the VDA (for example, by using Set Default Programs), that change can interfere with host to client redirection for applications.

When host to client content redirection is enabled, the app that is used to open the URL depends on the configuration of the user device for both the URL type and the content type. For example:

- An HTTP URL with an HTML content type will open in the default web browser.
- An HTTP URL with a PDF content type might open in the default web browser, or it might open in another application.

This user device configuration is not controlled by host to client content redirection. If you do not control the configuration of the user device, consider using Flash redirection and multimedia redirection, rather than host to client content redirection.

The following URL types are opened locally through user devices when host to client redirection is enabled:

- HTTP (Hypertext Transfer Protocol)
- HTTPS (Secure Hypertext Transfer Protocol)
- RTSP (Real Player and QuickTime)
- RTSPU (Real Player and QuickTime)
- PNM (Legacy Real Player)
- MMS (Microsoft Media Format)

You can change the list of URL types for host to client redirection, to remove and add URL types, including custom URL types.

Enabling host to client redirection starts with enabling a Citrix policy setting.

The Host to client redirection policy setting is located in the [File Redirection policy settings](#) section. By default, this setting is disabled.

In addition, you may need to set registry keys and Group Policy for the server VDAs, depending on the VDA's OS.

- If the server VDA is Windows Server 2008 R2 SP1, you do not need to set registry keys or Group Policy.
- If the server VDA is Windows Server 2012, Windows Server 2012 R2, or Windows Server 2016, you must set registry keys and Group Policy.

警告

Using Registry Editor incorrectly can cause serious problems that might require you to reinstall your operating system. Citrix cannot guarantee that problems resulting from the incorrect use of Registry Editor can be solved. Use Registry Editor at your own risk. Be sure to back up the registry before you edit it.

1. Copy the text between "Reg file start" and "Reg file end" below, and paste it in Notepad.
2. Save the Notepad file with "Save As" as type All Files and the name ServerFTA.reg.
3. Distribute the **ServerFTA.reg** file to the servers using Active Directory Group Policy.

- Reg file start --

Windows Registry Editor Version 5.00

[HKEY_CLASSES_ROOT\ServerFTAHTML\shell\open\command]

@="\"C:\\Program Files (x86)\\Citrix\\system32\\iexplore.exe\" %1"

[HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\ServerFTA]

@="ServerFTA"

[HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\ServerFTA\Capabilities]

"ApplicationDescription"="Server FTA URL."

"ApplicationIcon"="C:\\Program Files (x86)\\Citrix\\system32\\iexplore.exe,0"

"ApplicationName"="ServerFTA"

[HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\ServerFTA\Capabilities\URLAssociations]

"http"="ServerFTAHTML"

"https"="ServerFTAHTML"

[HKEY_LOCAL_MACHINE\SOFTWARE\RegisteredApplications]

"Citrix.ServerFTA"="SOFTWARE\\Citrix\\ServerFTA\\Capabilities"

-- Reg file end --

Create an XML file. Copy the text between "xml file start" and "xml file end" below, paste it in the XML file, and then save the file as **ServerFTAdefaultPolicy.xml**.




```

-- xml file start --

<?xml version="1.0" encoding="UTF-8"?>

<DefaultAssociations>

<Association Identifier="http" ProgId="ServerFTAHTML" ApplicationName="ServerFTA" />

<Association Identifier="https" ProgId="ServerFTAHTML" ApplicationName="ServerFTA" />

</DefaultAssociations>

-- xml file end --

```

From the current Group Policy Management Console, navigate to: **Computer configuration > Administrative Templates > Windows Components > File Explorer > Set a default associations configuration file**, and provide the ServerFTAdefaultPolicy.xml file you created.

To change the list of URL types for host to client redirection, set the following registry key on the server VDA.

Key: HKLM\Software\Wow6432Node\Citrix\SFTA

To remove URL types from the list, set DisableServerFTA and NoRedirectClasses:

Name: DisableServerFTA

Type: REG_DWORD

Data: 1

Name: NoRedirectClasses

Type: REG_MULTI_SZ

Data: Specify any combination of the values: http, https, rtsp, rtspu, pnm, or mms. Enter multiple values on separate lines. For example:

http

https

rtsp

To add URL types to the list, set ExtraURLProtocols:

Name: ExtraURLProtocols

Type: REG_MULTI_SZ

Data: Specify any combination of URL types. Each URL type must include the :// suffix; separate multiple values with semicolons. For example:

customtype1://;customtype2://

To enable host to client redirection for a specific set of web sites, set the following registry key on the server VDA.

Key: HKLM\Software\Wow6432Node\Citrix\SFTA

Name: ValidSites

Type: REG_MULTI_SZ

Data: Specify any combination of fully-qualified domain names (FQDNs). Enter multiple FQDNs on separate lines. An FQDN may include a wildcard in the leftmost position only. This matches a single level of domain, which is consistent with the rules in RFC 6125. For example:

www.example.com

*.example.com

适用于 Windows 桌面操作系统的 GPU 加速

May 04, 2016

通过 HDX 3D Pro，可在桌面操作系统计算机上随托管桌面或应用程序交付图形密集型应用程序。HDX 3D Pro 支持物理主计算机（包括桌面、刀片式计算机和机架工作站）、具有 GPU 直通功能的 XenServer VM 以及具有虚拟 GPU (vGPU) 的 XenServer VM。

利用 XenServer GPU 直通功能，可以创建对专用图形处理硬件具有独占访问功能的 VM。可以在虚拟机管理程序上安装多个 GPU，并将 VM 一对一地分配给每个 GPU。

利用 XenServer vGPU，多个虚拟机可以直接访问单个物理 GPU 的图形处理功能。真正的硬件 GPU 共享功能可以为具有复杂和苛刻设计要求的用户提供完整而又合适的 Windows 7 或 Windows 2008 R2 SP1 桌面。GPU 共享功能适用于 NVIDIA GRID K1 和 K2 卡，使用非虚拟化操作系统上部署的相同 NVIDIA 图形驱动程序。

HDX 3D Pro 提供以下功能：

- 基于 H.264 的自适应深度压缩，用于实现最佳的 WAN 和无线性能。HDX 3D Pro 使用基于 CPU 的深度压缩作为编码的默认压缩技术。此技术提供了最佳压缩性能，可动态适应网络情况。

基于 H.264 的深度压缩编解码器不再与 NVIDIA GPU 上 CUDA 核心的图形呈现争用资源。深度压缩编解码器在 CPU 上运行并提供带宽效率。

- 专用的无损压缩选项。HDX 3D Pro 还提供基于 CPU 的无损编解码器，可支持需要在像素级完美呈现图形的应用程序，例如医学成像。建议仅针对特殊用例使用无损压缩，因为这种压缩方式占用相当多的网络和处理资源。

使用无损压缩时：

- 无损指示器（一个系统托盘图标）会通知用户显示的屏幕是有损帧还是无损帧。当视觉质量策略设置指定设为无损时，此功能很有用。当发送的是无损帧时，无损指示器将变绿。
- 无损切换功能使用户能够在会话内随时切换到“始终无损”模式。要在会话内随时选择或取消选择“无损”，请在该图标上单击鼠标右键或使用快捷键 Alt+Shift+1。对于无损压缩：HDX 3D Pro 使用无损编解码器进行压缩，而不考虑通过策略选择的编解码器。

对于有损压缩：HDX 3D Pro 使用原始编解码器，即默认编解码器或通过策略选择的编解码器。

后续会话不会保留无损转换设置。要为每个连接使用无损编解码器，请在视觉质量策略设置中选择始终无损。

- 在 **7.6 FP3** 中，可以覆盖用于在会话内选择或取消选择无损的默认快捷方式 ALT+SHIFT+1。在 HKLM\SOFTWARE\Citrix\HDX3D\LLIndicator 配置新注册表设置。
 - 名称：HKLM_HotKey，类型：String
 - 配置快捷键组合的格式为 C=0|1, A=0|1, S=0|1, W=0|1, K=val。注册表项必须使用逗号 (,) 分隔。按键顺序无关紧要。
 - A、C、S、W 和 K 表示按键，其中 C=Control、A=ALT、S=SHIFT、W=Win 和 K=某个有效按键。K 允许的值包括 0-9、a-z 和所有虚拟键代码。有关虚拟键代码的详细信息，请参阅 MSDN 上的 [Virtual-Key Codes](#)（虚拟键代码）。
 - 例如：
 - 对于 F10，设置 K=0x79
 - 对于 Ctrl + F10，设置 C=1、K=0x79
 - 对于 Alt + A，设置 A=1、K=a 或 A=1、K=A 或 K=A、A=1
 - 对于 Ctrl + Alt + 5，设置 C=1、A=1、K=5 或 A=1、K=5、C=1
 - 对于 Ctrl + Shift + F5，设置 A=1、S=1、K=0x74

警告：注册表编辑不当会导致严重问题，可能导致需要重新安装操作系统。Citrix 无法保证因“注册表编辑器”使用不当导致出现的问题能够得以解决。使用“注册表编辑器”需自担风险。在编辑注册表之前，请务必进行备份。

- 多监视器和高分辨率监视器支持。对于 Windows 7 和 Windows 8 桌面，HDX 3D Pro 支持最多配有四个显示器的用户设备。用户可以采用任意配置安排自己的显示器，并且可以混合使用分辨率和方向各不相同的显示器。显示器的数量受主机计算机 GPU 功能、用户设备以及可用带宽限制。HDX 3D Pro 支持所有显示器分辨率，并仅受主机计算机上 GPU 的功能限制。

HDX 3D Pro 还对双显示器访问 Windows XP 桌面提供有限支持。有关详细信息，请参阅[运行 Windows XP 或 Windows Vista 的计算机上的 VDA](#)。

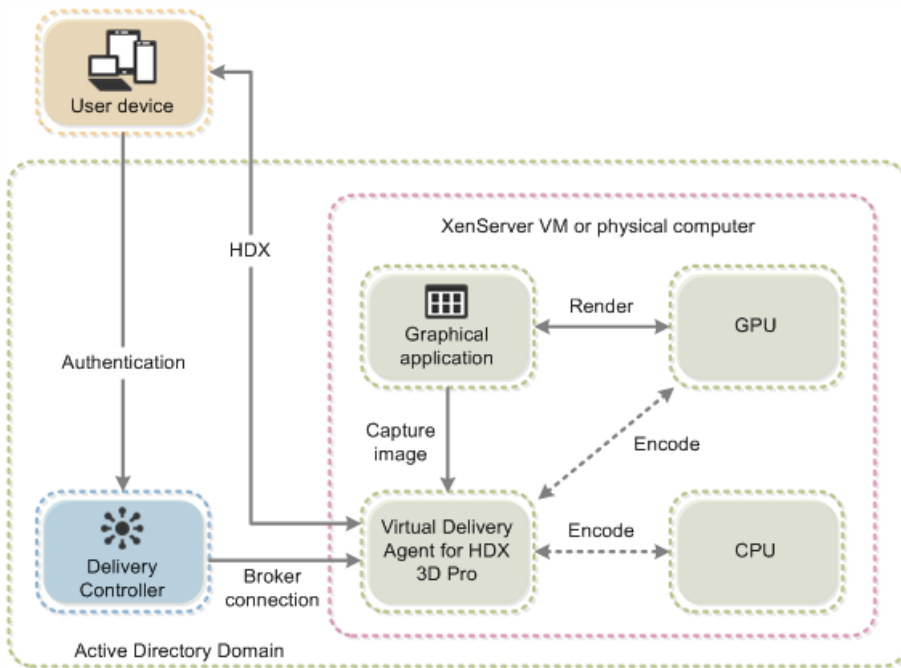
- 动态分辨率。可以将虚拟桌面或应用程序窗口的分辨率调整为任意大小。**注意：**唯一受支持的更改分辨率的方法为调整 VDA 会话窗口的大小。不支持从 VDA 会话内部更改分辨率（使用“控制面板”>“外观和个性化”>“显示”>“屏幕分辨率”）。
- 支持 NVIDIA Kepler 体系结构。HDX 3D Pro 支持 NVIDIA GRID K1 和 K2 卡，以便实现 GPU 直通和 GPU 共享。NVIDIA GRID vGPU 允许多个 VM 使用在非虚拟化操作系统中部署的相同 NVIDIA 图形驱动程序同时直接访问单个物理 GPU。
- 支持使用虚拟直接图形加速 (vDGA) 的 VMware vSphere 和 VMware ESX — 可针对 RDS 和 VDI 工作负载将 HDX 3D Pro 与 vDGA 结合使用。当 HDX 3D Pro 与 Virtual Shared Graphics Acceleration (vSGA) 结合使用时，仅支持一个显示器。将 vSGA 用于大型 3D 模型会导致性能问题，因为它使用 API 截获技术。有关详细信息，请参阅 [VMware vSphere 5.1 - Citrix 已知问题](#)。

如下图所示：

- 主机计算机必须与 Delivery Controller 驻留在相同的 Active Directory 域中。
- 用户登录到 Citrix Receiver 并访问虚拟应用程序或桌面时，控制器将对用户进行身份验证，并与 VDA for HDX 3D Pro 建立连接，以将连接转至托管图形应用程序的计算机。

VDA for HDX 3D Pro 使用主机上相应的硬件来压缩完整桌面的视图或仅压缩图形应用程序的视图。

- 此桌面或应用程序视图以及用户与这些视图之间的交互将通过 Citrix Receiver 与 VDA for HDX 3D Pro 之间的直接 HDX 连接在主机计算机与用户设备之间传输。



使用安装程序的图形界面安装 VDA for Windows Desktop OS 时，只需在 HDX 3D Pro 页面选择是。使用命令行接口时，请在 XenDesktop VdaSetup.exe 命令中包含 /enable_hdx_3d_pro 选项。

要升级 HDX 3D Pro，应先卸载单独的 HDX 3D for Professional Graphics 组件以及 VDA，然后再安装 VDA for HDX 3D Pro。同样，要从标准 VDA for Windows Desktop OS 切换到 HDX 3D Pro VDA，请先卸载标准 VDA，然后再安装 VDA for HDX 3D Pro。

NVIDIA GRID API 提供了对 GPU 帧缓冲区的直接访问，为实现流畅的交互式用户体验提供了尽可能最快的帧速率。如果您在安装 VDA with HDX 3D Pro 前安装 NVIDIA 驱动程序，则默认启用 NVIDIA GRID。

要在 VM 上启用 NVIDIA GRID，请从 Device Manager 禁用 Microsoft 基本显示适配器。运行以下命令，然后重新启动 VDA：Montereyenable.exe -enable -noreset

如果您在安装 VDA with HDX 3D Pro 后安装 NVIDIA 驱动程序，NVIDIA GRID 将禁用。使用 NVIDIA 提供的 Montereyenable 工具启用 NVIDIA GRID。

要禁用 NVIDIA GRID，请运行以下命令，然后重新启动 VDA：Montereyenable.exe -disable -noreset

要将 HDX 3D Pro 用于多个显示器，请确保主机计算机已配置的显示器数不少于连接到用户设备的显示器数。连接到主机计算机的显示器可以是物理机，也可以是虚拟机。

在用户连接到提供图形应用程序的虚拟桌面或应用程序时，禁止将显示器（无论是物理机还是虚拟机）连接到主机计算机。否则，会引起用户会话持续时间的不稳定。

请告诉用户，图形应用程序会话运行期间，不支持（由用户或应用程序）对桌面分辨率进行更改。关闭应用程序会话后，用户可以在“Citrix Receiver - Desktop Viewer 首选项”中更改 Desktop Viewer 窗口的分辨率。

多位用户共享一个带宽有限的连接时（例如，在分支机构），Citrix 建议您使用总会话带宽限制策略，以限制每位用户可用的带宽。这样可确保用户登录和注销时可用带宽不会大幅波动。由于 HDX 3D Pro 可自动调整以充分利用所有可用带宽，因此，在用户会话过程中可用带宽大幅波动可能会对性能产生负面影响。

例如，如果 20 位用户共享一个 60 Mbps 的连接，每位用户可用的带宽可能在 3 Mbps 到 60 Mbps 之间变化，具体取决于并发用户的数量。要优化此种情形下的用户体验，应确定高峰时段每位用户所需的带宽，并将用户限制为始终使用此带宽量。

对于 3D 鼠标用户，Citrix 建议将通用 USB 重定向虚拟通道的优先级提高到 0。有关更改虚拟通道优先级的信息，请参阅 [CTX128190](#)。

Windows Server 操作系统 GPU 加速

May 04, 2016

通过 HDX 3D Pro，在 Windows Server 操作系统会话中运行的图形密集型应用程序可以在服务器的图形处理器 (GPU) 上呈现。通过将 OpenGL、DirectX、Direct3D 和 Windows Presentation Foundation (WPF) 呈现移到服务器的 GPU 上，服务器的 CPU 不会因图形呈现而变慢。此外，服务器还能够处理更多图形，因为工作负载在 CPU 和 GPU 之间进行了拆分。

使用 HDX 3D Pro 时，多个用户可以共享图形卡。当 HDX 3D Pro 与 XenServer GPU 直通结合使用时，单台服务器可托管多个图形卡，每个虚拟机托管一个。

有关涉及到编辑注册表的过程，请注意：注册表编辑不当会导致严重问题，可能需要重新安装操作系统。Citrix 无法保证因“注册表编辑器”使用不当导致出现的问题能够得以解决。使用“注册表编辑器”需自担风险。在编辑注册表之前，请务必进行备份。

GPU Sharing 使 GPU 硬件可以在远程桌面会话中呈现 OpenGL 和 DirectX 应用程序；它具有以下特点：

- 可用于裸机或虚拟机，以提高应用程序的可扩展性及性能。
- 启用多个并发会话以共享 GPU 资源（大多数用户并不需要专用 GPU 的呈现性能）。
- 无需任何特殊设置。

可以在虚拟机管理程序上安装多个 GPU，并将 VM 一对一地分配给每个 GPU：方法是安装一个配备有多个 GPU 的图形卡，或者安装多个分别配备一个或多个 GPU 的图形卡。建议不要在服务器上混合使用异类图形卡。

虚拟机需要直接传递访问 GPU，这可通过使用 Citrix XenServer 或 VMware vSphere 实现。当 HDX 3D Pro 与 GPU 直通结合使用时，服务器中的每个 GPU 支持一台多用户虚拟机。

GPU Sharing 不依赖任何特定的图形卡。

- 在虚拟机管理程序上运行时，请选择与虚拟机管理程序的 GPU 直通实现兼容的硬件平台和图形卡。有关已通过 XenServer GPU 直通证书测试的硬件列表，请访问 [GPU 直通设备](#)。
- 在裸机上运行时，建议使用操作系统启用的一个显示适配器。如果在硬件上安装了多个 GPU，请仅保留一个 GPU，并使用 Device Manager 禁用其余的 GPU。

使用 GPU Sharing 的可扩展性取决于多个因素：

- 正在运行的应用程序
- 占用的视频 RAM 量
- 图形卡的处理能力

例如，对于运行 ESRI ArcGIS 等应用程序的 NVIDIA Q6000 和 M2070Q 卡，已报告具有 8 至 10 位用户的可扩展性。这些卡提供 6 GB 的视频 RAM。较新的 NVIDIA GRID 卡提供 8 GB 的视频 RAM 和显著提高的处理能力（更多 CUDA 核心）。使用 NVIDIA GRID K2 卡时，每个 GRID K2 卡支持 20 用户时可实现良好的性能。其他应用程序可能会提高更多，在高端 GPU 上达到 32 个并发用户。

一些应用程序处理视频 RAM 短缺的能力要优于其他应用程序。如果硬件严重过载，可能会导致图形卡驱动程序不稳定或崩溃。可限制并发用户的数量，以避免此类问题。

可以使用第三方工具（如 GPU-Z）来确定是否已实现 GPU 加速。GPU-Z 可从 <http://www.techpowerup.com/gpuz/> 获取。

DirectX、Direct3D 和 WPF 呈现仅在具有支持显示驱动程序接口 (DDI) 9ex、10 或 11 版的 GPU 的服务器上可用。

- 在 Windows Server 2008 R2 上，DirectX 和 Direct3D 不需要特殊设置即可使用单个 GPU。
- 在 Windows Server 2012 上，RD 会话主机服务器上的远程桌面服务 (RDS) 会话将 Microsoft 基本呈现驱动程序用作默认适配器。要在 Windows Server 2012 上的 RDS 会话中使用 GPU，请启用组策略“本地计算机策略”>“计算机配置”>“管理模板”>“Windows 组件”>“远程桌面服务”>“远程桌面会话主机”>“远程会话环境”中的对所有远程桌面服务会话使用硬件默认图形适配器。
- 要能够使用服务器的 GPU 呈现 WPF 应用程序，请在运行 Windows 服务器操作系统会话的服务器的注册表中创建以下设置：
 - [HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CtxHook\Applnit_Dlls\Multiple Monitor Hook]
"EnableWPFHook"=dword:00000001
 - [HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\CtxHook\Applnit_Dlls\Multiple Monitor Hook]
"EnableWPFHook"=dword:00000001

支持对用户会话中运行的 CUDA 和 OpenCL 应用程序进行 GPU 加速（实验阶段）。默认情况下，会禁用此支持功能，但可以启用该功能以进行测试和评估。

要使用 CUDA 加速功能（实验阶段），请启用以下注册表设置：

- [HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CtxHook\Applnit_Dlls\Graphics Helper] "CUDA"=dword:00000001
- [HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\CtxHook\Applnit_Dlls\Graphics Helper]
"CUDA"=dword:00000001

要使用 OpenCL 加速功能（实验阶段），请启用以下注册表设置：

- [HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CtxHook\Applnit_Dlls\Graphics Helper] "OpenCL"=dword:00000001
- [HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\CtxHook\Applnit_Dlls\Graphics Helper]
"OpenCL"=dword:00000001

OpenGL Software Accelerator

May 04, 2016

OpenGL Software Accelerator 是一个适用于 OpenGL 应用程序（例如 ArcGIS、Google Earth、Nehe、Maya、Blender、Voxler、CAD 和 CAM）的软件光栅器。在某些情况下，借助 OpenGL Software Accelerator，无需使用图形卡即可在 OpenGL 应用程序中提供极佳的用户体验。

重要：OpenGL Software Accelerator 按“原样”提供，必须使用所有应用程序进行测试。OpenGL Software Accelerator 可能不适用于某些应用程序，旨在解决 Windows OpenGL 光栅器性能低下的问题。如果 OpenGL Software Accelerator 与应用程序结合使用，可用来避免 GPU 硬件的成本。

OpenGL Software Accelerator 在安装介质的 Support（支持）文件夹中提供，在所有的有效 VDA 平台上受支持。

何时应尝试使用 OpenGL Software Accelerator？

- 如果 XenServer 或其他虚拟机管理程序上的虚拟机中运行的应用程序存在性能问题，可尝试使用 OpenGL Accelerator。对某些应用程序而言，OpenGL Accelerator 优于 Windows 中的 Microsoft OpenGL 软件光栅器，因为 OpenGL Accelerator 采用 SSE4.1 和 AVX。OpenGL Accelerator 还支持使用 OpenGL 2.1 及之前版本的应用程序。
- 对于在工作站上运行的应用程序，请首先尝试使用工作站图形适配器所提供的默认 OpenGL 支持功能版本。如果图形卡是最新版本，则大多数情况下它将提供最佳性能。如果图形卡是旧版本或者无法提供令人满意的性能，则可以尝试使用 OpenGL Software Accelerator。
- 对于在使用基于 CPU 的软件光栅器时无法充分发挥功能的 3D OpenGL 应用程序，可能有必要使用 OpenGL GPU 硬件加速功能。此功能可用于裸机或虚拟机。

音频功能

May 04, 2016

可以向某个策略配置并添加以下 Citrix 策略设置以优化 HDX 音频功能。有关详细用法以及与其他策略设置的关系和相关性，请参阅[音频策略设置](#)、[带宽策略设置](#)和[多流连接策略设置](#)。

重要：大多数音频功能都通过 ICA 流进行传输，这些功能受安全保护的方式与其他 ICA 通信流相同。未安装 NetScaler Access Gateway 时，用户数据报协议 (User Datagram Protocol, UDP) 音频使用独立的未受安全保护的传输机制。如果 NetScaler Access Gateway 配置为访问 XenApp 和 XenDesktop 资源，则端点设备与 NetScaler Access Gateway 之间的音频流量将使用 DTLS 协议确保安全。

通常情况下，音频质量越高，需要向用户设备发送的音频数据就越多，占用的带宽也就越多，服务器 CPU 使用率也就越高。借助声音压缩功能，可以在音频质量与整体会话性能之间取得平衡。可使用 Citrix 策略设置来配置要应用于声音文件的压缩级别。

默认情况下，音频质量策略设置为高 - 高清晰度音频。此设置可提供高保真立体声音频，但占用的带宽高于其他质量设置。对于未优化的语音聊天或视频聊天应用程序（如软件电话），请不要使用此音频质量，因为此音频质量可能会在音频路径中产生延迟，不适合用于实时通信。

考虑为拨号用户组和通过 LAN 或 WAN 连接的用户组创建单独的策略。如果使用通常限制带宽的拨号连接，对用户而言，下载速度比声音质量更加重要。因此，您可能需要为拨号连接和 LAN 或 WAN 连接分别创建单独的策略，前者所应用的音频压缩级别较高，后者所应用的音频压缩级别较低。

有关设置的详细信息，请参阅[音频策略设置](#)。请务必在用户设备上启用客户端自动设置；请参阅[用户设备的音频设置策略](#)。

要允许用户在用户设备上通过扬声器或其他音频设备（如耳机）从服务器上的应用程序接收音频，请添加客户端音频重定向设置，此设置的默认值为允许。

客户端音频映射可能导致服务器和网络负载过大；但是，禁止客户端音频重定向将禁用所有 HDX 音频功能。

有关设置的详细信息，请参阅[音频策略设置](#)。请务必在用户设备上启用客户端自动设置；请参阅[用户设备的音频设置策略](#)。

要允许用户使用用户设备上的麦克风等输入设备录制音频，请添加客户端麦克风重定向设置，此设置的默认值为允许。

出于安全考虑，当不受用户设备信任的服务器尝试访问麦克风时，系统会向用户发出警报。用户可以在使用麦克风之前选择接受或拒绝访问。用户可以在 Citrix Receiver 上禁用此警报功能。

有关设置的详细信息，请参阅[音频策略设置](#)。请务必在用户设备上启用客户端自动设置；请参阅[用户设备的音频设置策略](#)。

音频即插即用策略设置可控制是否允许使用多个音频设备来录制和播放声音。默认情况下，此设置处于启用状态。

此设置仅适用于 Windows 服务器操作系统计算机。

有关设置的详细信息，请参阅[音频策略设置](#)。

音频重定向带宽限制策略设置指定在会话中播放和录制音频时所用的最大带宽 (Kbps)。音频重定向带宽限制百分比设置指定音频重定向功能所用的最大带宽占总会话带宽的百分比。默认情况下，这两项设置指定为零（无最大值）。如果同时配置了这两个设置，则使用最低带宽限制的那个设置。

有关设置的详细信息，请参阅[带宽策略设置](#)。请务必在用户设备上启用客户端自动设置；请参阅[用户设备的音频设置策略](#)。

默认情况下，通过 UDP 协议的音频实时传输设置为允许（如果在安装过程中选择），从而在服务器上打开一个 UDP 端口，以支持使用“通过 UDP 协议的音频实时传输”的连接。Citrix 建议为音频配置 UDP/RTP 协议，以确保在发生网络拥堵或数据包丢失时获得最佳的用户体验。

重要：如果未安装 NetScaler Access Gateway，则不加密通过 UDP 传输的音频数据。如果 NetScaler Access Gateway 配置为访问 XenApp 和 XenDesktop 资源，则端点设备与 NetScaler Access Gateway 之间的音频流量将使用 DTLS 协议确保安全。

音频 UDP 端口范围指定 Virtual Delivery Agent (VDA) 用来与用户设备交换音频数据包数据的端口号范围。

默认情况下，范围为 16500 - 16509。

有关通过 UDP 协议的音频实时传输的设置详细信息，请参阅[音频策略设置](#)；有关音频 UDP 端口范围的详细信息，请参阅[多流连接策略设置](#)。请务必在用户设备上启用客户端自动设置；请参阅[用户设备的音频设置策略](#)。

1. 按照[使用组策略对象模板配置 Receiver](#)进行操作，加载组策略模板。
2. 在组策略编辑器中，依次展开“管理模板”>“Citrix 组件”>“Citrix Receiver”>“用户体验”。
3. 对于客户端音频设置，请选择未配置、启用或禁用。
 - **未配置。** 默认情况下，通过高质量音频或以前配置的自定义音频设置启用音频重定向。
 - **已启用。** 音频重定向通过选定的选项启用。
 - **已禁用。** 音频重定向已禁用。
4. 如果选择已启用，请选择声音质量。对于 UDP 音频，请仅使用中（默认设置）。
5. （仅适用于 UDP 音频）选择**启用实时传输**，然后设置用于在本地 Windows 防火墙中打开的传入端口的范围。
6. 要通过 NetScaler Access Gateway 使用 UDP 音频，请选择**允许通过网关实时传输**。应为 NetScaler Access Gateway 配置 DTLS。有关详细信息，请参阅[通过 Netscaler Gateway 传输 UDP 音频](#)。

作为管理员，如果您在端点设备上没有控制权限，无法进行更改（例如，BYOD 或使用家用计算机时），请使用 StoreFront 中的 default.ica 属性启用 UDP 音频。

1. 在 StoreFront 计算机上，使用编辑器（例如记事本）打开 C:\inetpub\wwwroot\Citrix\App_Data\default.ica。
2. 将以下条目移至 [Application] 部分下。



```
; This is to enable Real-Time Transport

EnableRtpAudio=true

; This is to Allow Real-Time Transport Through gateway

EnableUDPThroughGateway=true

; This is to set audio quality to Medium

AudioBandwidthLimit=1 -

; UDP Port range

RtpAudioLowestPort=16500

RtpAudioHighestPort=16509
```

如果通过编辑 default.ica 启用 UDP 音频，则 UDP 音频将对使用该存储的所有用户启用。

用户参与音频或视频会议时可能会听到回声。通常当扬声器和麦克风彼此间距离太近的时候会产生回声。因此，Citrix 建议在音频和视频会议中使用耳机。

HDX 提供了一个回声消除选项（默认情况下处于启用状态），可以将回声降低到最小。扬声器和麦克风之间的距离直接影响回声消除功能的效果；这些设备之间的距离不宜过近或过远。

您可以更改注册表设置以禁用回声消除功能。在注册表中操作时请注意：注册表编辑不当会导致严重问题，可能需要重新安装操作系统。Citrix 无法保证因“注册表编辑器”使用不当导致出现的问题能够得以解决。使用“注册表编辑器”需自担风险。在编辑注册表之前，请务必进行备份。

1. 在用户设备上使用注册表编辑器导航到以下位置：

- 32 位计算机：HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\ICA
Client\Engine\Configuration\Advanced\Modules\ClientAudio\EchoCancellation
- 64 位计算机：HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\ICA
Client\Engine\Configuration\Advanced\Modules\ClientAudio\EchoCancellation

2. 将值数据字段更改为 FALSE。

网络流量优先级

May 04, 2016

对于使用支持服务质量 (QoS) 的路由器的会话，可以跨多个连接为网络流量分配优先级。可使用四个 TCP/IP 流（实时、交互、后台和批量通信）和一个 UDP/RTP 流（语音通信）在用户设备与服务器之间传输 ICA 通信。每个虚拟通道都有一个特定的优先级，并通过相应连接进行传输。可以根据连接所使用的 TCP 端口号分别设置这些通道。

对于安装在 Windows 8 和 Windows 7 计算机上的 Virtual Delivery Agent (VDA)，支持多通道流连接。请与贵公司的网络管理员协作，确保在多端口策略设置中配置的通用网关协议 (CGP) 端口已正确分配到网络路由器。

仅在已配置多会话可靠性端口或 CGP 端口时，才支持服务质量 (QoS)。

警告：使用此功能时，请启用传输安全性。Citrix 建议使用 Internet 协议安全性 (Internet Protocol Security, IPsec) 或安全套接字层 (Secure Sockets Layer, SSL)。仅当连接在支持多流功能的 NetScaler Gateway 上进行遍历时，才支持 SSL 连接。在内部企业网络上时，不支持采用 SSL 的多流连接。

要为多流连接设置服务质量，请在策略中添加以下 Citrix 策略设置（有关详细信息，请参阅[多流连接策略设置](#)）：

- 多端口策略 – 此设置为跨多个连接的 ICA 通信指定端口，并确定网络优先级。
 - 从 CGP 默认端口优先级列表选择一个优先级；默认情况下，主端口 (2598) 的优先级为“高”。
 - 根据需要在 CGP 端口 1、CGP 端口 2 和 CGP 端口 3 中键入其他 CGP 端口，并标识每个端口的优先级。每个端口必须有唯一的优先级。

将 VDA 上的防火墙显式地配置为允许其他 TCP 流量。

- 多流计算机设置 – 默认情况下禁用此设置。如果要在环境中使用具有“多流”支持功能的 Citrix CloudBridge，则无需配置此设置。如果要使用第三方路由器或旧版 Branch Repeater 实现所需的服务质量 (QoS)，应配置此策略。
- 多流用户设置 – 默认情况下禁用此设置。

要使包含这些设置的策略生效，用户必须注销后再登录到网络。

USB 和客户端设备注意事项

May 04, 2016

利用 HDX USB 设备重定向，用户可以将闪存驱动器连接到本地计算机，并从虚拟桌面或桌面托管应用程序远程访问该驱动器。在会话期间，用户可以使用即插即用设备，包括图片传输协议 (PTP) 设备（如数码相机）、媒体传输协议 (MTP) 设备（如数字音频播放器或便携式媒体播放器）以及销售点 (POS) 设备。

桌面托管应用程序会话不支持双跳 USB。

USB 重定向适用于 Receiver for Windows 和 Receiver for Linux。

默认情况下，会允许某些类型的 USB 设备使用 USB 重定向功能，但拒绝其他类型的 USB 设备使用；请参阅 Receiver 文档了解详细信息。您可以通过更新支持重定向功能的 USB 设备的列表来限制可用于虚拟桌面的 USB 设备类型。

Important

对于需要将用户设备和服务器安全分离的环境，请向用户提供关于应避免使用的 USB 设备类型的指导信息。

经过优化的虚拟通道可用于重定向最常用的 USB 设备，并可以通过 WAN 提供性能和带宽效率。所提供的支持级别取决于用户设备上安装的 Receiver。通常情况下，经过优化的虚拟通道即是最佳选择，对于存在高延迟的环境更是如此。

为了执行 USB 重定向，此产品将以处理鼠标的相同方式来处理 SMART 板。

此产品支持对 USB 3.0 设备和 USB 3.0 端口使用优化的虚拟通道，例如，用于查看摄像机上的文件或向耳麦提供音频的 CDM 虚拟通道。此产品还支持连接到 USB 2.0 端口的 USB 3.0 设备的通用 USB 重定向。

对于没有优化虚拟通道的特殊设备，将退为使用通用的 USB 虚拟通道，该通道可提供原始的 USB 重定向功能。有关已在 XenDesktop 中经过测试的 USB 设备的信息，请参阅 [CTX123569](#)。

一些特定于设备的高级功能，如网络摄像机上的人体学接口设备 (HID) 按钮，在优化的虚拟通道中可能无法按预期运行；如果发生此问题，请使用通用 USB 虚拟通道。

默认情况下不会重定向某些设备，这些设备只能用于本地会话。例如，不应通过对内部 USB 连接到用户设备系统板的网络接口卡进行重定向。

下列 Citrix 策略设置可控制 USB 支持功能：

- **客户端 USB 设备优化规则。** 在使用 GPMC 的 7.6 FP3 中可用。类=03 的输入设备支持此优化模式，例如签名设备和手写平板。如果未指定任何规则，将按照交互模式 (02) 处理设备。捕获模式 (04) 是签名设备的建议模式。
- **客户端 USB 设备重定向。** 默认设置为禁止。
- **客户端 USB 设备重定向规则。** 规则仅适用于使用通用 USB 重定向的设备；因此，这些规则不适用于使用专用或优化重定向的设备，例如 CDM。
- **客户端 USB 即插即用设备重定向。** 默认情况下设置为允许，即允许在用户会话中使用即插即用的 PTP、MTP 和 POS 设备。
- **客户端 USB 设备重定向带宽限制。** 默认设置为 0（无最大值）。
- **客户端 USB 设备重定向带宽限制百分比。** 默认设置为 0（无最大值）。

通用 USB 重定向适用于没有优化的虚拟通道的专用 USB 设备。此功能可将任意 USB 设备从客户端计算机重定向到虚拟桌面；利用此功能，最终用户可以在桌面会话中选择与多种通用 USB 设备交互，其效果与物理连接这些设备一样。

利用通用 USB 重定向：

- 用户不需要在其设备上安装设备驱动程序
- USB 客户端驱动程序安装在 VDA 计算机上

从 VDA for Desktop OS 7.6 运行的桌面会话支持此功能。

从 VDA for Server OS 7.6 运行的桌面会话也支持此功能，但具有以下限制：

- VDA 计算机必须运行 Windows Server 2012 R2
- 仅支持单跳场景
- USB 客户端驱动程序必须与适用于 Windows 2012 R2 的 RDSH 兼容
- 不支持 USB 存储设备、音频设备、智能卡读卡器以及未完全虚拟化的设备

有关配置通用 USB 重定向的详细信息，请参阅 [CTX137939](#)。

1. 向策略中添加客户端 USB 设备重定向设置，并将其值设置为允许。
2. （可选）要更新可进行远程连接的 USB 设备的列表，请向策略中添加客户端 USB 设备重定向规则设置并指定 USB 策略规则。
3. 在用户设备上安装 Receiver 时启用 USB 支持。如果已在上一节中为 Virtual Delivery Agent 指定 USB 策略规则，请为 Receiver 指定相同的策略规则。对于瘦客户端，请向制造商咨询有关 USB 支持以及任何所需配置的详细信息。

当已启用 USB 支持并将 USB 用户首选项设置配置为自动连接 USB 设备时，将自动重定向 USB 设备。当在桌面设备模式下运行并且不存在连接栏时，也会自动重定向 USB 设备。但在某些情况下，您可能不希望自动重定向所有 USB 设备。有关详细信息，请参阅 [CTX123015](#)。

用户可以明确地对未自动重定向的设备执行重定向操作，方法是从 USB 设备列表中选择这些设备。要阻止列出或重定向 USB 设备，可以在客户端和 VDA 上指定设备规则，如下文中所述。

- 可以通过同时为 Receiver 和 VDA 指定 USB 设备重定向规则以覆盖默认 USB 策略规则，来更新可进行远程连接的 USB 设备范围。
- 编辑用户设备注册表。安装介质中包含一个管理模板（ADM 文件），以便您可以通过 Active Directory 组策略更改用户设备：dvd root \os\lang\Support\Configuration\icaclient_usb.adm。
 - 通过组策略规则为服务器操作系统计算机编辑管理员覆盖规则。组策略管理控制台包含在安装介质上：
 - 对于 x64：dvd root \os\lang\x64\Citrix Policy\ CitrixGroupPolicyManagement_x64.msi
 - 对于 x86：dvd root \os\lang\x86\Citrix Policy\ CitrixGroupPolicyManagement_x86.msi

警告

注册表编辑不当会导致严重问题，可能导致需要重新安装操作系统。Citrix 无法保证因“注册表编辑器”使用不当导致出现的问题能够得以解决。使用“注册表编辑器”需自担风险。在编辑注册表之前，请务必进行备份。

产品的默认规则存储在 HKLM\SOFTWARE\Citrix\PortICA\GenericUSB\DeviceRules 中。请勿编辑这些产品默认规则，而应根据这些规则创建管理员覆盖规则，如下文中所述。组策略对象覆盖规则将在产品默认规则之前进行评估。

管理员覆盖规则存储在 HKLM\SOFTWARE\Policies\Citrix\PortICA\GenericUSB\DeviceRules 中。组策略对象策略规则的格式为 {Allow:|Deny;} 后接一组以空格分隔的 tag=value 表达式。支持以下标记：

标记	说明
VID	设备描述符中的供应商 ID
PID	设备描述符中的产品 ID
REL	设备描述符中的版本 ID
Class	设备描述符或接口描述符中的类；请参阅 USB Web 站点 http://www.usb.org/ 了解可用的 USB 类代码
SubClass	设备描述符或接口描述符中的子类
Prot	设备描述符或接口描述符中的协议

- 创建新策略规则时，应注意以下事项：
- 规则不区分大小写。
 - 规则末尾可以带有以 # 开头的可选注释。无需分隔符，且将忽略注释以使规则匹配。
 - 空白注释行和纯注释行会被忽略。
 - 空格用作分隔符，但不能出现在数字或标识符中间。例如，Deny: Class = 08 SubClass=05 是有效规则，Deny: Class=0 Sub Class=05 则无效。
 - 标识必须使用匹配运算符 =。例如，VID=1230。
 - 每条规则都必须另起新行，或包含在以分号分隔的列表中。

Important

如果使用 ADM 模板文件，则必须在一行中创建规则（以分号分隔的列表）。

使用优化设备（例如大容量存储设备）时，通常使用专用 CDM 通道而非策略规则重定向设备。但是，可以通过以下方式之一覆盖此行为：

- 使用“通用 USB 重定向”手动重定向优化的设备：从首选项对话框的设备选项卡中选择 切换到通用。
- 使用“通用 USB 重定向”自动重定向优化的设备：设置存储设备的自动重定向（例如 AutoRedirectStorage = 1），并将 USB 用户首选项设置为自动连接 USB 设备；有关详细信息，请参阅 [CTX123015](#)。

- 示例：
- 以下示例显示了一个用于供应商和产品标识符的 USB 策略规则，由管理员定义：
Allow: VID=046D PID=C626 # Allow Logitech SpaceNavigator 3D Mouse Deny: VID=046D # Deny all Logitech products
 - 以下示例显示了一个用于已定义类、子类和协议的 USB 策略规则，由管理员定义：
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

默认情况下，当已启用 USB 支持并将 USB 用户首选项设置配置为自动连接 USB 设备时，将自动对 USB 设备执行重定向操作。此外，对于桌面设备站点或桌面托管应用程序，也将自动重定向 USB 设备。但在某些情况下，您可能不希望自动重定向所有 USB 设备。有关详细信息，请参阅 [CTX123015](#)。

Desktop Viewer 用户可以对未自动重定向的设备执行重定向操作，方法是从 USB 设备列表中选择这些设备。要阻止列出或重定向 USB 设备，可以在用户设备和 VDA 上指定设备规则。

- 可以通过同时为 Receiver 和 VDA 指定 USB 设备重定向规则以覆盖默认 USB 策略规则，来更新可进行远程连接的 USB 设备范围。设备规则将同时对 Receiver 和 VDA 强制实施。请确保同时执行更改，以便设备远程连接可以如您期望的那样进行。
- 编辑用户设备的注册表（对于 Receiver for Linux，则应编辑 .ini 文件）。安装介质中包含一个管理模板（ADM 文件），以便您可以通过 Active Directory 组策略更改用户设备：dvd root

os\lang\Support\Configuration\icaclient_usb.adm。

- 在服务器操作系统计算机上的 VDA 注册表中编辑管理员覆盖规则。安装介质中包含一个 ADM 文件，以便您通过 Active Directory 组策略更改 VDA：dvd root os\lang\Support\Configuration\vda_usb.adm。

警告

注册表编辑不当会导致严重问题，可能导致需要重新安装操作系统。Citrix 无法保证因“注册表编辑器”使用不当导致出现的问题能够得以解决。使用“注册表编辑器”需自担风险。在编辑注册表之前，请务必进行备份。

产品的默认规则存储在 HKLM\SOFTWARE\Citrix\PortICA\GenericUSB\DeviceRules 中。请勿编辑这些产品默认规则，而应根据这些规则创建管理员覆盖规则，如下文中所述。组策略对象覆盖规则将在产品默认规则之前进行评估。

管理员覆盖规则存储在 HKLM\SOFTWARE\Policies\Citrix\PortICA\GenericUSB\DeviceRules 中。组策略对象策略规则的格式为 {Allow:|Deny:} 后接一组以空格分隔的 tag=value 表达式。支持以下标记：

标记	说明
VID	设备描述符中的供应商 ID
PID	设备描述符中的产品 ID
REL	设备描述符中的版本 ID
Class	设备描述符或接口描述符中的类；请参阅 USB Web 站点 http://www.usb.org/ 了解可用的 USB 类代码
SubClass	设备描述符或接口描述符中的子类
Prot	设备描述符或接口描述符中的协议

创建新策略规则时，应注意以下事项：

- 规则不区分大小写。
- 规则末尾可以带有以 # 开头的可选注释。无需分隔符，且将忽略注释以使规则匹配。
- 空白注释行和纯注释行会被忽略。
- 空格用作分隔符，但不能出现在数字或标识符中间。例如，Deny: Class = 08 SubClass=05 是有效规则，Deny: Class=0 Sub Class=05 则无效。
- 标识必须使用匹配运算符 =。例如，VID=1230。
- 每条规则都必须另起新行，或包含在以分号分隔的列表中。

Important

如果使用 ADM 模板文件，则必须在一行中创建规则（以分号分隔的列表）。

使用优化设备（例如大容量存储设备）时，通常使用专用 CDM 通道而非策略规则重定向设备。但是，可以通过以下方式之一覆盖此行为：

- 使用“通用 USB 重定向”手动重定向优化的设备：从首选项对话框的设备选项卡中选择 切换到通用。
- 使用“通用 USB 重定向”自动重定向优化的设备：设置存储设备的自动重定向（例如 AutoRedirectStorage = 1），并将 USB 用户首选项设置为自动连接 USB 设备；有关详细信息，请参阅 [CTX123015](#)。

示例：

- 以下示例显示了一个用于供应商和产品标识符的 USB 策略规则，由管理员定义：
Allow: VID=046D PID=C626 # Allow Logitech SpaceNavigator 3D Mouse Deny: VID=046D # Deny all Logitech products
- 以下示例显示了一个用于已定义类、子类和协议的 USB 策略规则，由管理员定义：
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

用户可以在启动虚拟会话之前或之后连接 USB 设备。

当使用 Receiver for Windows 时，以下情况适用：

- 在会话启动后连接的设备将立即显示在 Desktop Viewer 的 USB 菜单中。
- 如果 USB 设备不能正确重定向，可等到虚拟会话启动后再连接设备，这样可以解决此问题。
- 为避免数据丢失，请使用 Windows 的“安全删除硬件”图标来移除 USB 设备。

远程访问可以通过客户端驱动器映射（即用户登录时，用户设备上的驱动器将自动映射至虚拟桌面上的驱动器盘符）实现，这仅适用于大容量存储设备。这些驱动器显示为具有映射驱动器盘符的共享文件夹。要配置客户端驱动器映射，请使用 ICA 策略设置的“文件重定向策略设置”部分中的客户端可移动驱动器设置。

两种类型的远程连接策略之间的主要区别如下：

功能	客户端驱动器映射	通用 USB 重定向
默认情况下处于启用状态	是	否
可配置只读访问权限	是	否
可在会话期间安全删除设备	否	是，只要用户按照操作系统建议进行安全删除

如果同时启用了通用 USB 和客户端驱动器映射策略，并且在会话启动之前或之后插入了大容量存储设备，则将使用客户端驱动器映射对其进行重定向。如果同时启用了通用 USB 和客户端驱动器映射策略，设备配置为自动重定向（请参见 <http://support.citrix.com/article/CTX123015>），并且在会话启动之前或之后插入了大容量存储设备，则将使用通用 USB 对其进行重定向。

注意

低带宽连接（如 50 Kbps）条件下支持 USB 重定向，但是复制大型文件不起作用。

您可以控制用户是否能够将文件从虚拟环境复制到用户设备。默认情况下，可在读取/写入模式下从会话内部使用映射的客户端驱动器上的文件和文件夹。

要防止用户添加或修改映射的客户端设备上的文件和文件夹，请启用只读客户端驱动器访问策略设置。将此设置添加到策略中时，请确保客户端驱动器重定向设置为允许并且也已添加到策略。

监视

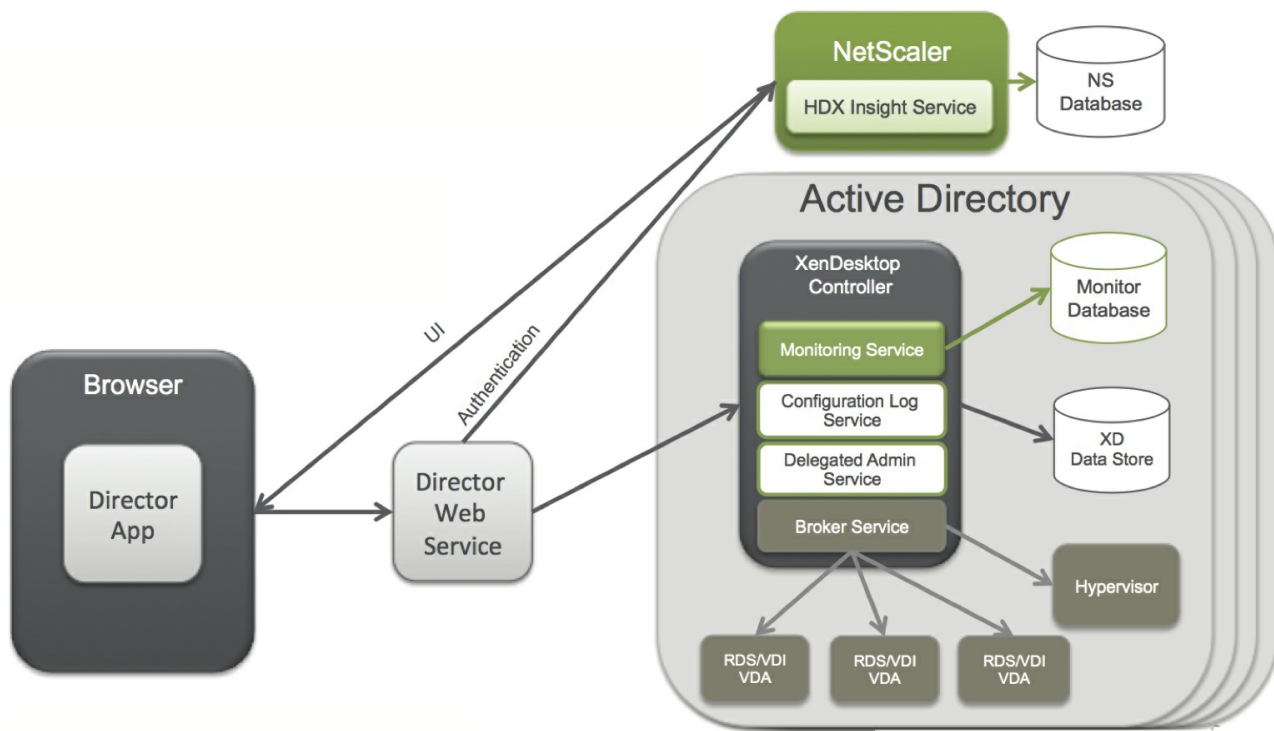
May 04, 2016

管理员和技术支持人员可以利用 Director 监视 XenApp 和 XenDesktop 站点，在这种情况下，管理员可以访问配置日志记录数据库或使用站点的监视服务 API（使用 OData 协议）。

管理员可以监视：

- 会话使用情况
- 登录性能
- 连接和计算机故障
- 负载评估
- 历史趋势
- 基础结构
- 用户会话
- 计算机

Director 是一款实时 Web 工具，管理员可以利用此工具进行监视、排除故障和为用户执行支持任务。



Director 可以访问：

- 使用集成 EdgeSight 功能、Performance Manager 和 Network Inspector 的统一控制台访问来自 Broker Agent 的实时数据。
 - EdgeSight 功能包括用于运行状况、容量保障、历史趋势和网络分析的性能管理，由 NetScaler HDX Insight 提供技术支持，可识别由 XenApp 或 XenDesktop 环境中的网络导致的瓶颈。
- 存储在监视数据库中的历史数据，用于访问配置日志记录数据库。

- 使用 HDX Insight 访问 NetScaler Gateway 中的 ICA 数据。
 - 可以了解 XenApp 或 XenDesktop 的虚拟应用程序、桌面和用户的最终用户体验。
 - 将网络数据与应用程序数据和实时指标关联起来，以便有效进行故障排除。
 - 与 XenDesktop 7 Director 监视工具集成。
- 允许运行时监视的个人虚拟磁盘数据显示基本分配，并使技术支持 IT 人员能够重置个人虚拟磁盘（重置为仅在必要时使用）。
 - 命令行工具 CtxPvdDiag.exe 用于将用户日志信息收集到一个文件中，以供执行故障排除使用。

Director 使用故障排除控制板。此控制板提供对 XenApp 或 XenDesktop 站点的实时运行状况监视。利用此功能，管理员可以实时查看故障，更好地了解最终用户的体验。

通过 Session Recording，您可以在遵守公司政策和法规的前提下，从运行 XenApp 的任意服务器通过任意类型的连接录制任意用户会话的屏幕活动。Session Recording 对会话进行记录、编录和存档，以便进行检索和播放。

Session Recording 使用灵活的策略自动触发应用程序会话的录制。这样就可以使 IT 能够监视和检查应用程序（如金融运营和医疗患者信息系统）的用户活动，在遵守法规和安全监视方面加强内部控制。同样，Session Recording 也通过加快问题识别及问题解决时间在技术支持方面予以帮助。

利用配置日志记录功能，管理员可以跟踪对 XenApp 或 XenDesktop 站点所做的管理更改。配置日志记录可以帮助管理员诊断和排除配置更改后出现的问题，辅助进行变更管理和跟踪配置，并报告管理活动。

可以在 Director 中使用 Trend View 界面查看配置日志记录，以便向不具有 XenDesktop Citrix Studio 访问权限的管理员提供配置更改通知。

Trends View 提供一段时间内的配置更改历史数据，使管理员可以访问对站点所做的更改、更改时间和执行更改的人员，以便查找问题的原因。此视图将配置信息细分为三类。

- 连接故障
- 出现故障的桌面计算机
- 出现故障的服务器计算机

管理员可以使用站点的监视服务 API（使用 OData 协议）搜索历史数据。这使得 IT 部门可以分析历史趋势以进行规划；详尽排除连接故障和计算机故障；提取信息以供其他工具和进程所用。

Monitor Service 架构提供以下类型的数据：

- 与连接故障相关的数据
- 处于故障状态的计算机
- 会话使用情况
- 登录持续时间
- 负载平衡数据

- [Director](#)
- [Session Recording - 适用于 XenApp 7.6 FP1 和 FP2](#)
- [监视个人虚拟磁盘](#)

- [配置日志记录](#)
- [Monitor Service OData API](#)

Director

May 04, 2016

Director 提供了面向特定管理员定制的不同界面视图。产品权限决定显示的内容和可用的命令。

例如，技术支持管理员可以看到专为技术支持任务定制的界面。Director 允许技术支持管理员搜索报告问题的用户并显示该用户相关的活动，例如用户的应用程序和进程的状态。他们可以通过执行相应操作来快速解决问题，例如终止无响应的应用程序或进程，重影用户计算机上的操作，重新启动计算机或重置用户配置文件。

相比之下，完全权限管理员可以查看和管理整个站点，并且可以对多个用户和计算机执行命令。控制板提供了部署各主要方面的概况，例如会话状态、用户登录和站点基础结构。信息每分钟更新一次。如果出现问题，将会自动显示有关所发生故障的数量和类型的详细信息。

默认情况下，Director 作为 Web 站点安装在 Delivery Controller 上。有关必备条件和其他详细信息，请参阅此版本的“系统要求”文档。

此版本的 Director 与 6.5 版本之前的 XenApp 部署或 7 版本之前的 XenDesktop 部署不兼容。

当在包含多个站点的环境中使用 Director 时，请确保对安装了 Controller、Director 和其他核心组件的所有服务器上的系统时钟进行同步。否则，站点可能无法在 Director 中正确显示。

提示：如果您计划监视 XenApp 6.5 和 XenApp 7.5 或 XenDesktop 7.x 站点，Citrix 建议将 Director 与用于监视 XenApp 6.5 站点的 Director 控制台安装在不同的服务器上。

重要：要保护通过网络使用纯文本发送的用户名和密码，Citrix 强烈建议您仅允许使用 HTTPS 进行 Director 连接，请勿使用 HTTP。某些工具可以读取 HTTP（未加密）网络数据包中的纯文本用户名和密码，这会对用户造成安全风险。

配置权限

要登录 Director，具有 Director 权限的管理员必须是 Active Directory 域用户并且必须具有以下权限：

- 对要搜索的所有 Active Directory 林的读取权限（请参阅[高级配置](#)）
- 配置的委派管理员角色（请参阅[委派管理和 Director](#)）。
- 要重影用户，必须使用适用于 Windows 远程协助的 Microsoft 组策略来配置管理员。此外：
 - 安装 VDA 时，确保在所有用户设备（默认处于选中状态）上启用 Windows 远程协助功能。
 - 在服务器上安装 Director 时，确保已安装 Windows 远程协助（默认处于选中状态）。但默认情况下服务器上禁用此功能。无需对 Director 启用此功能，即可为最终用户提供协助。Citrix 建议将此功能保持禁用状态，以提高服务器的安全性。
 - 要使管理员能够启动 Windows 远程协助，请使用远程协助的相应 Microsoft 组策略设置向其授予所需的权限。有关信息，请参阅 [CTX127388: How to Enable Remote Assistance for Desktop Director](#)（如何为 Desktop Director 启用远程协助）。
- 对于安装了版本 7 之前的 VDA 的用户设备，必须执行附加配置。请参阅[为 XenDesktop 7 之前版本的 VDA 配置权限](#)。

安装 Director

注意：要允许 Director 查找场中的所有 XenApp 工作组，需要在此场使用的 DNS 服务器上为 XenApp 服务器所在的子网添加反向 DNS 区域。

使用安装程序安装 Director，该安装程序将检查必备条件，安装任何缺少的组件，设置 DirectorWeb 站点以及执行基本配置。安装程序提供的默认配置可处理典型部署。如果在安装期间未安装 Director，请使用安装程序添加 Director。要添加任何其他

组件，请重新运行安装程序并选择要安装的组件。有关使用安装程序的信息，请参阅“安装”文档。Citrix 建议仅使用产品安装程序进行安装，请勿使用 MSI 文件。

当 Director 安装在 Controller 上时，将自动配置 localhost 作为服务器地址，并且默认情况下，Director 将与本地 Controller 进行通信。

要在 Controller 的远程专用服务器上安装 Director，系统将提示您输入 Controller 的 FQDN 或 IP 地址。默认情况下，Director 与指定的 Controller 进行通信。仅为要监视的每个站点指定一个 Controller 地址。Director 将自动发现同一站点中的所有其他 Controller，并且如果您指定的 Controller 出现故障，则将回退到其他 Controller。

注意：Director 无法在 Controller 之间平衡负载。

要确保浏览器与 Web 服务器之间的通信安全，Citrix 建议您在托管 Director 的 IIS Web 站点上实施 SSL。有关说明，请参阅 Microsoft IIS 文档。无需对 Director 执行任何配置即可启用 SSL。

安装 Director 7.6.300

Director 7.6.300 提供对 Framehawk 虚拟通道的支持并提供最新的产品修复。它随 XenApp 和 XenDesktop 7.6 FP2 以及 XenApp 6.5 FP3 提供。

注意：确保已在 IIS 中选中所有必需的功能。有关完整列表，请参阅 [CTX142260](#)。

1. 下载 Director，运行 MSI 文件 DesktopDirector.MSI 或 DesktopDirector_x64.MSI。
2. 安装 CitrixGroupPolicyManagement.MSI，此文件位于 XenApp 和 XenDesktop 安装介质上的 Citrix Policy（Citrix 策略）文件夹中。
3. 为 Director 配置 Delivery Controller，使用 C:\inetpub\wwwroot\Director\tools 中提供的 DirectorConfig.exe 工具。有关详细信息，请参阅 [CTX137990](#)。
4. 向 IIS 注册 ASP.net。为此，请运行命令 C:\inetpub\wwwroot\Director\tools>DirectorConfig.exe /registerdotnet
5. 在 VDA 上安装或升级 WMIPProxy，然后重新启动计算机。WMIPProxy_x64.MSI 和 WMIPProxy_x86.MSI 包含在 Director 7.6.300 下载文件中。注意：如果不升级 WMIPProxy，Framehawk 虚拟通道将显示为与启用 Framehawk 的 VDA“不兼容”。
6. 在 Delivery Controller 上升级 XDPoshSnapin_Hotfix，然后重新启动 Studio。XDPoshSnapin MSI 位于第 1 步下载的 Director 文件中。委派管理员和自定义管理员要想查看 Framehawk 虚拟通道信息，需要进行此升级。

登录 Director

DirectorWeb 站点位于 https 或 http:///Director。

如果多站点部署中的一个站点出现故障，在尝试连接到该故障站点时，登录 Director 所需的时间会稍长。

安装 Director for XenApp 6.5

如果已为 XenDesktop 安装 Director，请按如下所示完成 XenApp 的配置：

- 在每个 Director 服务器上，使用 IIS 管理器控制台在应用程序设置中更新 XenApp 服务器地址的列表，如[高级配置](#)中“向 Director 中添加站点”部分所述。按 XenApp 场提供一个控制器的服务器地址：XenApp 场中的所有其他控制器将自动用于故障转移。Director 无法在 Controller 之间平衡负载。
- 将每台 XenApp 工作服务器配置为接受 WinRM 查询，如[配置权限](#)中所述。
- 为端口 2513 配置防火墙例外，Director 与 XenApp 之间进行通信时使用此端口。

首次安装 Director for XenApp 6.5

首次安装 Director for XenApp 6.5 时，请执行以下步骤。通常，将 Director 与 XenApp 控制器安装在不同的计算机上。

1. 从 XenApp 7.6 安装介质安装 Director
2. 在每个 Director 服务器上，使用 IIS 管理器控制台在应用程序设置中更新 XenApp 服务器地址的列表，如[高级配置](#)中“向 Director 中添加站点”部分所述。
按 XenApp 场提供一个控制器的服务器地址：XenApp 场中的所有其他控制器自动用于故障转移。Director 无法在 Controller 之间平衡负载。

重要：对于 XenApp 地址，请确保使用设置 Service.AutoDiscoveryAddressesXA，而非默认设置 Service.AutoDiscoveryAddresses。

3. Director WMI 提供程序的安装程序位于 DVD 上的 Support\DirectorWMIPProvider 文件夹中。将其安装在所有合适的 XenApp 服务器（运行会话的控制器和工作服务器）上。
如果未配置 winrm，请运行 winrm qc 命令。
4. 将每台 XenApp 工作服务器配置为接受 WinRM 查询，如[配置权限](#)中所述。
5. 为端口 2513 配置防火墙例外，Director 与 XenApp 之间进行通信时使用此端口。
6. 要确保浏览器与 Web 服务器之间的通信安全，Citrix 建议您在托管 Director 的 IISWeb 站点上实现 SSL。
有关说明，请参阅 Microsoft IIS 文档。无需对 Director 执行任何配置即可启用 SSL。

委派管理和 Director

May 04, 2016

委派管理基于三个概念：管理员、角色和作用域。权限基于管理员的角色以及该角色的作用域。例如，可以为管理员指派技术支持管理员角色，其作用域只包括负责一个站点上的最终用户。

有关创建委派管理员的信息，请参阅[委派管理](#)文档。

管理权限决定着向管理员呈现的 Director 界面以及他们可以执行的任务。权限决定着以下事项：

- 用户可以访问的页面，统称为视图。
- 管理员可以查看并与之交互的桌面、计算机和会话。
- 管理员可以执行的命令，例如重影用户的会话或启用维护模式。

内置角色和权限还决定着管理员对 Director 的使用方式：

管理员角色	在 Director 中的权限
完全权限管理员	对所有视图具有完全访问权限，并且可以执行所有命令，包括重影用户的会话、启用维护模式和导出趋势数据。
交付组管理员	对所有视图具有完全访问权限，并且可以执行所有命令，包括重影用户的会话、启用维护模式和导出趋势数据。
只读权限管理员	可以访问所有视图并查看指定作用域中的所有对象，还可以查看全局信息。可以从 HDX 通道下载报告，并且可以使用“趋势”视图中的“导出”选项导出趋势数据。 无法执行任何其他命令或在视图中进行任何更改。
技术支持管理员	只可以访问“技术支持”和“用户详细信息”视图，并且只可以查看委派管理员进行管理的对象。可以重影用户会话并为该用户执行命令。可以执行维护模式操作。可以对桌面操作系统计算机使用电源控制选项。 无法访问控制板、“趋势”或“过滤器”视图。无法对服务器操作系统计算机使用电源控制选项。
计算机目录管理员	无访问权限。Director 不支持此管理员，因此其无法查看数据。此用户可以访问“计算机详细信息”页面（基于计算机的搜索）。
主机管理员	无访问权限。Director 不支持此管理员，因此其无法查看数据。

配置 Director 管理员的自定义角色

在 Studio 中，还可以配置 Director 特定的自定义角色，以更好地满足组织的需求，并且更灵活地委派权限。例如，您可以限制内置的技术支持管理员角色，使该管理员无法从会话注销。

如果通过 Director 权限创建一个自定义角色，还必须向该角色分配其他通用权限：

- 用于登录 Director 的 Delivery Controller 权限。

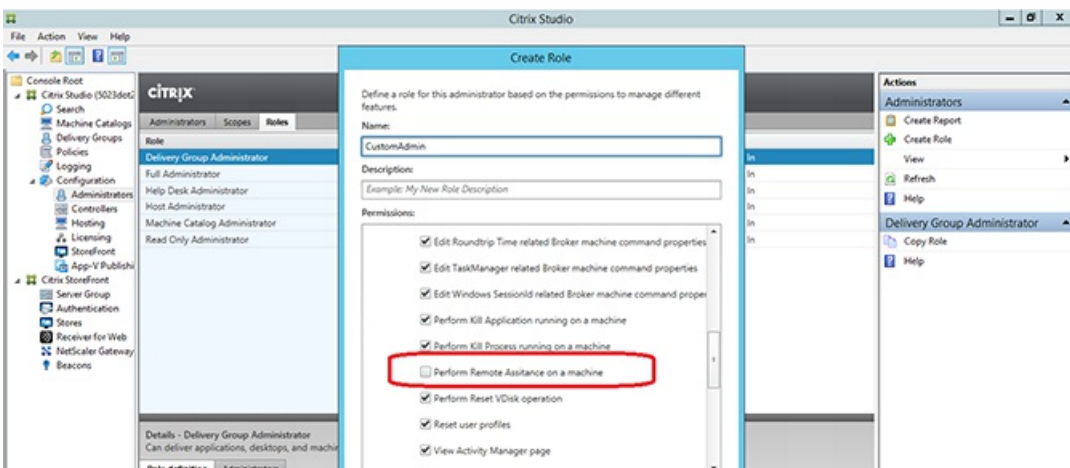
- 用于查看与 Director 中的交付组相关的数据的交付组权限。

此外，还可以通过复制现有角色创建自定义角色，并包括不同视图的附加权限。例如，可以复制技术支持角色并包括用于查看“控制板”或“过滤器”页面的权限。

为自定义角色选择 Director 权限，包括：

- 在计算机上执行终止应用程序的操作
- 在计算机上执行终止进程的操作
- 在计算机上执行远程协助
- 执行重置虚拟磁盘操作
- 重置用户配置文件
- 查看“客户端详细信息”页面
- 查看“控制板”页
- 查看“过滤器”页面
- 查看“计算机详细信息”页面
- 查看“趋势”页面
- 查看“用户详细信息”页

在此示例中，重影（在计算机上执行远程协助）关闭。



另外，从其他组件的权限列表中，考虑选择以下权限：

- 从交付组：
 - 使用交付组成员身份启用/禁用计算机的维护模式
 - 使用交付组成员身份在 Windows 桌面计算机上执行电源操作
 - 使用交付组成员身份在计算机上执行会话管理

为 XenDesktop 7 之前版本的 VDA 配置权限

May 04, 2016

如果用户在其设备上安装了 XenDesktop 7 之前的 VDA 版本，Director 会通过 Windows 远程管理 (WinRM) 从部署中补充有关实时状态和指标的信息。

此外，使用此过程配置 WinRM，使其能与 XenDesktop 5.6 Feature Pack1 中的 Remote PC 兼容使用。

默认情况下，只有桌面计算机的本地管理员（通常为域管理员及其他特权用户）才具有查看实时数据所需的权限。

有关安装并配置 WinRM 的信息，请参阅 [CTX125243](#)。

要使用户能够查看实时数据，必须向其授予相应权限。例如，假定有多个 Director 用户（HelpDeskUserA、HelpDeskUserB 等等）同属于名为 HelpDeskUsers 的 Active Directory 安全组。已在 Studio 中为该组指定了技术支持管理员角色，从而为这些用户提供了必需的 Delivery Controller 权限。但该组还需要具有从桌面计算机访问实时数据所需的权限。

要提供所需的访问权限，可以通过以下两种方式之一配置必需的权限：

- 向 Director 用户授予权限（模拟模式）
- 向 Director 服务授予权限（可信子系统模式）

向 Director 用户授予权限（模拟模式）

默认情况下，Director 使用模拟模式：WinRM 与桌面计算机之间通过 Director 用户身份进行连接。因此，Desktop Director 用户必须在该台式机上具有相应的权限。

可以通过以下两种方式（本文档稍后会介绍）之一配置这些权限：

1. 将用户添加到桌面计算机的本地管理员组中。
2. 向用户授予 Director 所需的特定权限。此选项避免了授予 Director 用户（例如，HelpDeskUsers 组）计算机上的完全管理权限。

向 Director 服务授予权限（可信子系统模式）

可以对 Director 进行配置，使 WinRM 连接使用服务标识，并仅向该服务标识授予相应的权限，而无需向 Director 用户授予对桌面计算机的相应权限。

在此模式下，Director 用户本身将无权执行 WinRM 调用。他们可以使用 Director 访问数据。

IIS 中的 Director 应用程序池配置为以服务标识方式运行。默认情况下，这是 APPPOOL\DesktopDirector 虚拟帐户。执行远程连接时，此帐户将显示为服务器的 Active Directory 计算机帐户，例如 MyDomain\DirectorServer\$。必须将此帐户配置为具有相应权限。

如果部署了多个 Director Web 站点，则必须将每个 Web 服务器的计算机帐户都置于配置了相应权限的 Active Directory 安全组中。

要将 Director 设置为使用 WinRM 的服务标识而非用户身份，请按[高级配置](#)中所述配置以下设置：

Service.Connector.WinRM.Identity = Service

可以通过以下两种方式之一配置这些权限：

1. 将服务帐户添加到桌面计算机上的本地管理员组中。

2. 向服务帐户授予 Director 所需的特定权限（如下文介绍）。此选项可避免向服务帐户授予计算机的完全管理权限。

向特定用户或用户组分配权限

要使 Director 能够通过 WinRM 从桌面计算机访问所需的信息，需要具有以下权限：

- WinRM RootSDDL 中的读取和执行权限
- WMI 命名空间权限：
 - root/cimv2 – 远程访问权限
 - root/citrix – 远程访问权限
 - root/RSOP – 远程访问权限和执行权限
- 这些本地组的成员关系：
 - 性能监视用户
 - 事件日志读者

用于自动授予上述权限的 ConfigRemoteMgmt.exe 工具分别位于 x86\Virtual Desktop Agent 文件夹和 x64\Virtual Desktop Agent 文件夹中的安装介质上以及 tools 文件夹中的安装介质上。必须向所有 Director 用户授予上述权限。

要向 Active Directory 安全组、用户和计算机帐户授予这些权限，或授予执行结束应用程序和结束进程操作的权限，使用管理权限从命令提示窗口运行此工具，并在运行时采用以下参数：

ConfigRemoteMgmt.exe /configwinrmuser domain\name

其中 name 为安全组、用户或计算机帐户。

要向某个用户安全组授予所需的权限，请运行：

ConfigRemoteMgmt.exe /configwinrmuser domain\HelpDeskUsers

要向特定计算机帐户授予权限，请运行：

ConfigRemoteMgmt.exe /configwinrmuser domain\DirectorServer\$

对于结束进程、结束应用程序和重影操作：

ConfigRemoteMgmt.exe /configwinrmuser domain\name /all

要向某个用户组授予权限，请运行：

ConfigRemoteMgmt.exe /configwinrmuser domain\HelpDeskUsers /all

要显示工具的帮助，请运行：

ConfigRemoteMgmt.exe

配置 HDX Insight

May 04, 2016

注意：此功能的可用性取决于组织的许可证和管理员权限。

HDX Insight 将 EdgeSight 网络分析和 EdgeSight Performance Management 与 Director 相集成：

- EdgeSight 网络分析利用 HDX Insight 来提供应用程序和桌面的网络上下文视图。使用此功能，Director 可对其部署中的 ICA 通信进行高级分析。
- EdgeSight Performance Management 可提供历史保留和趋势报告。通过历史数据保留与实时评估，可以创建趋势报告，其中包括容量趋势和运行状况趋势。

在 Director 中启用此功能后，HDX Insight 可为 Director 提供更多信息：

- 趋势页可以显示对整个部署的应用程序、桌面和用户的延迟和带宽影响。
- 用户详细信息页可以显示特定于某个特殊用户会话的延迟和带宽信息。

限制

- ICA 会话的往返程时间 (RTT) 可正确显示 Receiver for Windows 3.4 或更高版本以及 Receiver for Mac 11.8 或更高版本的数据。对于早期版本的 Receiver，数据无法正确显示。
- 在“趋势”视图中，不会针对早于版本 7 的 VDA 收集 HDX 连接登录数据。对于更早版本的 VDA，图表数据将显示为 0。

在 Director 上配置 EdgeSight 网络分析功能

EdgeSight 可通过利用 NetScaler HDX Insight 提供网络分析功能，以向 Citrix 应用程序和桌面的管理员提供对可能导致网络性能不佳的问题进行故障排除和关联的功能。

必须在 Director 中安装并配置 NetScaler Insight Center，才能启用 EdgeSight 网络分析。Insight Center 是一种虚拟机（设备），可从 Citrix.com 下载。通过使用 EdgeSight 网络分析，Director 可以传送和收集与部署相关的信息。这些信息可在 HDX Insight 中进行利用，这让 HDX Insight 可以对客户端和后端 Citrix 基础结构之间的 Citrix ICA 协议进行可靠分析。

1. 在安装 Director 的服务器上，在 C:\inetpub\wwwroot\Director\tools 中找到 DirectorConfig 命令行工具，并在命令行提示窗口中使用参数 /confignetscaler 运行该工具。
2. 系统弹出提示时，请按提示配置 NetScaler Insight Center 计算机名称（FQDN 或 IP 地址）、用户名、密码和 HTTP 或 HTTPS 连接类型。
3. 要验证更改，请先注销，然后再重新登录。

高级配置

May 04, 2016

在本文中：

[支持跨多个 Active Directory 域和林的用户](#)

[向 Director 中添加站点](#)

[在活动管理器中禁止显示运行中的应用程序](#)

某些高级 Director 配置（例如，支持多个站点或多个 Active Directory 林）通过 Internet Information Services (IIS) 管理器中的设置进行控制。

重要：如果更改了 IIS 中的某项设置，Director 服务会自动重新启动并注销用户。

使用 IIS 配置高级设置：

1. 打开 Internet Information Services (IIS) 管理器模块。
2. 转到默认 Web 站点下的 Director Web 站点。
3. 双击**应用程序设置**。
4. 双击某个设置以对其进行编辑。

默认情况下，铂金版许可证保留 90 天以内的数据。有关配置的详细信息，请参阅[数据粒度和保留](#)。

Director 使用 Active Directory 搜索用户并查找其他用户和计算机信息。默认情况下，Director 搜索：

- 管理员帐户所属的域或林。
- Director Web 服务器所属的域或林（如果不相同）。

Director 将尝试使用 Active Directory 全局目录在林级别执行搜索。如果管理员没有相应的权限，无法在林级别执行搜索，则仅搜索域。

要搜索或查询其他 Active Directory 域或林中的数据，必须明确设置要搜索的域或林。配置以下设置：

`Connector.ActiveDirectory.Domains = (user),(server)`

值属性 user 和 server 分别代表 Director 用户（管理员）的域和 Director 服务器的域。

要使用户能够从其他域或林中进行搜索，请将该域的名称添加到列表中，如下例中所示：

`Connector.ActiveDirectory.Domains = (user),(server),,`

对于列表中的每个域，Director 将尝试在林级别执行搜索。如果管理员没有相应的权限，无法在林级别执行搜索，则仅搜索域。

如果已安装 Director，可将其配置为使用多个站点。要执行此操作，请在每个 Director 服务器上使用 IIS 管理器控制台来更新应用程序设置中服务器地址的列表。

将每个站点中的 Controller 地址添加到以下设置中：

`Service.AutoDiscoveryAddresses = SiteAController,SiteBController`

其中 SiteAController 和 SiteBController 为两个不同站点中的 Delivery Controller 地址。

对于 XenApp 6.5 站点，请将每个 XenApp 场中的控制器地址添加到以下设置中：
`Service.AutoDiscoveryAddressesXA = FarmAController,FarmBController`
其中 FarmAController 和 FarmBController 为两个不同场中的 XenApp 控制器的地址。

对于 XenApp 6.5 站点，从 XenApp 场添加控制器的另一种方法为：
`DirectorConfig.exe /xenapp FarmControllerName`

默认情况下，Director 中的活动管理器显示用户会话正在运行的所有应用程序的列表。对 Director 中的活动管理器功能具有访问权限的所有管理员均可以查看此信息。对于委派管理员角色，完全权限管理员、交付组管理员和技术支持管理员均可以查看此信息。

为保护用户及其正在运行的应用程序的隐私，您可以禁止“应用程序”选项卡列出正在运行的应用程序。

警告：注册表编辑不当会导致严重问题，可能导致需要重新安装操作系统。Citrix 无法保证因“注册表编辑器”使用不当导致出现的问题能够得以解决。使用“注册表编辑器”需自担风险。在编辑注册表之前，请务必进行备份。

1. 在 VDA 上，修改位于 `HKLM\Software\Citrix\Director\TaskManagerDataDisplayed` 的注册表项。默认情况下，该注册表项设置为 1。将值更改为 0，这表示在活动管理器中将不显示信息。
2. 在安装了 Director 的服务器上，修改用于控制正在运行的应用程序可见性的设置。默认情况下，该值为 true，表示允许应用程序选项卡显示正在运行的应用程序。将该值更改为 false，表示禁用其可见性。此选项仅影响 Director 中的活动管理器，不影响 VDA。

修改以下设置的值：

`UI.TaskManager.EnableApplications = false`

重要：要禁止查看运行中的应用程序，Citrix 建议进行上述两项更改，以确保在活动管理器中不显示这些数据。

监视部署

Jan 04, 2017

如果您具有完全权限管理员权限，在打开 Director 时，控制板将提供一个中央位置来监视站点的运行状况和使用情况。

如果当前没有故障或者在过去 60 分钟内没有发生故障，各个面板将保持折叠状态。发生故障时，将自动显示特定的故障面板。

注意：某些选项或功能可能不可用，具体取决于组织的许可证和您的管理员权限。

面板	说明
用户连接故障	过去 60 分钟内的连接故障。单击总数旁的类别可以查看该类型故障的指标。在相邻的表中，该数量按照交付组细分。
发生故障的桌面操作系统计算机或发生故障的服务器操作系统计算机	在过去 60 分钟内按照交付组细分的所有故障。按类型（包括无法启动、引导时卡住以及未注册）细分的故障。对于服务器操作系统计算机，故障还包括计算机达到最大负载。
许可状态	- 许可证服务器警报由许可证服务器发送，并且还显示解决警报所需的操作。 - Delivery Controller 警报由 Delivery Controller 发送，显示 Controller 看到的许可状态的详细信息。 可以在 Studio 中设置警报的阈值。 如果许可证服务器版本早于 11.12.1 且/或 Delivery Controller 早于 XenApp 7.6 或 XenDesktop 7.6，则不显示许可证服务器和/或 Delivery Controller 警报。
已连接的会话	过去 60 分钟所有交付组中已连接的会话。
平均登录时长	过去 60 分钟的登录数据。左侧的大数值表示该小时内的平均登录时长。 此平均值中不包括 XenDesktop 7.0 之前版本的 VDA 的登录数据。
基础结构	站点的主机、控制器和基础结构的运行状态。查看性能警报。 对于主机，通过来自 XenServer 或 VMware 的信息监视连接状态及 CPU、内存、带宽（网络使用率）和存储器（磁盘使用情况）的运行状况。 例如，可以将 XenCenter 配置为在某个托管服务器或虚拟机的 CPU、网络 I/O 或磁盘 I/O 超过特定阈值时生成性能警报。默认情况下，警报重复时间间隔是 60 分钟，但您也可以配置此时间间隔。有关详细信息，请参阅 XenServer 文档中的配置性能警报。

注意：如果未显示某特定指标的图标，则表明您所使用的主机类型不支持该指标。例如，不提供 System Center Virtual Machine Manager (SCVMM) 主机、AWS 和 CloudStack 的运行状况信息。

继续使用以下选项（见下文）对问题进行故障排除：

- 控制用户计算机电源

- 阻止与计算机连接

如果会话断开连接，它将继续处于活动状态，其应用程序仍会运行，但用户设备将不再与该服务器通信。

操作	说明
查看用户当前连接的计算机或会话	在活动管理器和用户详细信息视图中，查看用户当前连接的计算机或会话，以及该用户具有访问权限的所有计算机和会话的列表。要访问此列表，请单击用户标题栏中的会话切换程序图标。请参阅 还原会话 。
查看跨所有交付组的连接会话总数	从控制板的已连接的会话窗格中，查看最后 60 分钟内跨所有交付组的连接会话总数。然后单击较大的总数，打开过滤器视图，您可在其中根据所选交付组及跨交付组的范围和使用情况显示图形会话数据。
查看更长时间段内的数据	在趋势视图上，选择会话选项卡，逐级浏览更长时间段内连接和断开连接的会话更为具体的使用情况数据（即最后 60 分钟之前的会话总数）。要查看此信息，请单击查看历史趋势。

注意：如果用户设备运行旧版的 Virtual Delivery Agent (VDA)，例如版本 7 以前的 VDA，Director 将无法显示有关会话的完整信息。而是显示一条消息，指示无法从用户详细信息视图和活动管理器面板中获取信息。

在控制板上单击数字或从过滤器菜单选择一个预定义的过滤器时，过滤器视图将打开，并根据选择的计算机或故障类型显示数据。

无法编辑预定义过滤器，但是可以将其保存为自定义过滤器，然后再进行修改。此外，可以跨所有交付组创建计算机、连接和会话的自定义过滤视图。

1. 选择视图：

- 计算机 — 选择桌面操作系统计算机或服务器操作系统计算机。这些视图显示了已配置计算机的数量。“服务器操作系统计算机”选项卡包括负载评估器指数，如果将鼠标悬停在链接上，则会指示性能计数器的分布情况和会话计数的工具提示。
- 会话 — 还可以从“计算机”视图查看会话计数。
- 连接 — 按不同时间段显示的过滤连接，包括过去 60 分钟、过去 24 小时或过去 7 天。

2. 对于“故障方式”，选择条件。

3. 根据需要，对每个视图使用其他选项卡以完成过滤。

4. 根据需要，选择其他列以执行进一步的故障排除。

5. 保存并命名过滤器。

以后要打开过滤器，请从过滤器菜单中选择故障类型（计算机、会话或连接），然后选择已保存的过滤器。

6. 如有需要，对于“计算机”或“连接”视图，请为在过滤列表中选择的所有计算机使用电源控制。可以在 [Citrix Director 7.6 Failure Reasons Troubleshooting Guide](#)（Citrix Director 7.6 失败原因故障排除指南）中找到计算机和连接失败的失败原因以及建议的操作。

7. 对于“会话”视图，使用会话控制或消息发送选项。

继续使用以下选项（见下文）对问题进行故障排除：

- 控制用户计算机电源
- 阻止与计算机连接

“趋势”视图提供每个站点的会话、连接故障、计算机故障、登录性能和负载评估的历史趋势信息。要查找此信息，请单击“控制面板”或“过滤器”视图，然后单击“趋势”。

通过放大细节功能，管理员可以通过放大时间段（单击图标中的数据点）和不断细分来导航浏览趋势图，以查看与趋势关联的详细信息。借助此功能，您可以更好地详细了解所显示的趋势影响了哪些人员或哪些方面。

要更改每个图形的默认作用域，请对数据应用其他过滤器。

操作	说明
导出图形数据	选择包含要导出的数据的选项卡。单击“导出”，然后选择文件格式：.PDF、Excel 或 .CSV。
查看会话的趋势	从“会话”选项卡，选择“交付组”和时间段以查看关于并发会话计数的详细信息。
查看连接故障的趋势	从“连接故障”选项卡，选择计算机类型、故障类型、交付组和时间段，以查看包含有关站点间用户连接故障的更多详细信息的图形。
查看计算机故障的趋势	从“桌面操作系统计算机故障”选项卡或“服务器操作系统计算机”选项卡，选择故障类型、交付组和时间段，以查看包含有关站点间计算机故障的更多详细信息的图形。
查看登录性能的趋势	在“登录性能”选项卡中，选择交付组和时间段，以查看包含有关站点间用户登录次数的持续时间以及登录次数是否影响性能的更多详细信息的图形。该视图还显示了登录阶段的平均持续时间，例如代理时长、VM 开始时间等等。 此数据特别针对用户登录，不包括尝试重新连接已断开会话连接的用户。
查看负载评估的趋势	从“负载评估器指数”选项卡，查看包含负载在服务器操作系统计算机之间分发情况的更多详细信息的图表。此图表的过滤选项包括交付组或交付组中的服务器操作系统计算机、服务器操作系统计算机（仅当选择了交付组中的服务器操作系统计算机时才可用）和范围。
查看托管应用程序使用情况	此功能的可用性取决于组织的许可证。 在“托管应用程序使用情况”选项卡中，选择交付组和时间段，以查看一个显示最大并发使用情况的图和一个显示基于应用程序的使用情况的表。从“基于应用程序的使用情况”表格中，可以选择特定应用程序以查看详细信息和正在使用或曾经使用此应用程序的用户列表。
查看虚拟机使用情况	在“计算机使用情况”选项卡中，选择“桌面操作系统计算机”或“服务器操作系统计算机”以获取 VM 使用情况的实时视图，以便能够快速访问您的站点的容量需求。 桌面操作系统可用性 — 根据可用性显示整个站点或特定交付组的桌面操作系统计算机 (VDI) 的当前状态。 服务器操作系统可用性 — 根据可用性显示整个站点或特定交付组的服务器操作系统计算机 (VDI) 的当前状态。
使用 HDX Insight 查看网络分析数	此功能的可用性取决于组织的许可证和管理员权限。 在“网络”选项卡中，监视您的网络分析，其中提供网络的用户、应用程序和桌面上下文视图。通过此功能，Director 对部署中的 ICA 通信进行高级分析。

图表上的旗帜图标表示此特定时间范围内的重要事件或操作。将鼠标悬停在旗帜上并单击时，可列出事件或操作。

注意：对于版本 7 之前的 VDA 版本，不会收集 HDX 连接登录数据。对于更早版本的 VDA，图表数据将显示为 0。

继续使用以下选项（见下文）对问题进行故障排除：

- 控制用户计算机电源
- 阻止与计算机连接

趋势视图提供每个站点的会话、连接故障、计算机故障、登录性能和负载评估的历史趋势信息。要查找此信息，请单击控制板或过滤器视图，然后单击趋势。

要更改每个图形的默认作用域，请对数据应用其他过滤器。

操作	说明
导出图形数据	选择包含要导出的数据的选项卡。单击导出，然后选择文件格式：.PDF 或 .CSV。
查看会话的趋势	从会话选项卡，选择交付组和时间段以查看关于并发会话计数的详细信息。
查看连接故障的趋势	从连接故障选项卡，选择计算机类型、故障类型、交付组和时间段，以查看包含有关站点间用户连接故障的更多详细信息的图形。
查看计算机故障的趋势	从桌面操作系统计算机故障选项卡或服务器操作系统计算机选项卡，选择故障类型、交付组和时间段，以查看包含有关站点间计算机故障的更多详细信息的图形。
查看登录性能的趋势	在登录性能选项卡中，选择交付组和时间段，以查看包含有关站点间用户登录次数的持续时间以及登录次数是否影响性能的更多详细信息的图形。该视图还显示了登录阶段的平均持续时间，例如代理时长、VM 开始时间等等。 此数据特别针对用户登录，不包括尝试重新连接已断开会话连接的用户。
查看负载评估的趋势	从负载评估器指数选项卡，查看包含负载在服务器操作系统计算机之间分发情况的更多详细信息的图表。此图表的过滤选项包括交付组或交付组中的服务器操作系统计算机、服务器操作系统计算机（仅当选择了交付组中的服务器操作系统计算机时才可用）和范围。
查看托管应用程序使用情况	此功能的可用性取决于组织的许可证。 在托管应用程序使用情况选项卡中，选择交付组和时间段，以查看一个显示最大并发使用情况的图和一个显示基于应用程序的使用情况的表。从“基于应用程序的使用情况”表格中，可以选择特定应用程序以查看详细信息和正在使用或曾经使用此应用程序的用户列表。
使用 HDX Insight 查看网络分析数	此功能的可用性取决于组织的许可证和管理员权限。 在网络选项卡中，监视您的网络分析，其中提供网络的用户、应用程序和桌面上下文视图。通过此功能，Director 对部署中的 ICA 通信进行高级分析。

操作	说明

图表上的旗帜图标表示此特定时间范围内的重要事件或操作。将鼠标悬停在旗帜上并单击时，可列出事件或操作。

注意：对于版本 7 之前的 VDA 版本，不会收集 HDX 连接登录数据。对于更早版本的 VDA，图表数据将显示为 0。
继续使用以下选项（见下文）对问题进行故障排除：

- 控制用户计算机电源
- 阻止与计算机连接

要查看安装在特定计算机 VDA（物理或 VM）上的修补程序，请选择计算机详细信息视图。

要对您在 Director 中选择的计算机状态进行控制，请使用电源控制选项。这些选项对桌面操作系统计算机可用，但可能对服务器操作系统计算机不可用。

注意：对于物理机或使用 Remote PC Access 的计算机，此功能不可用。

命令	功能
重新启动	对 VM 执行顺序（软）关闭。在重新启动 VM 前，所有正在运行的进程将分别停止。例如，选择 Director 中显示为“启动失败”的计算机，并使用此命令重新启动这些计算机。
强制重新启动	在不预先执行任何关闭程序的情况下重新启动 VM。此命令与拔出然后插好物理服务器，并再次启动该服务器时作用相同。
关闭	对 VM 执行顺序（软）关闭；所有运行的进程将分别停止。
强制关闭	在不预先执行任何关闭程序的情况下关闭 VM。此命令与拔出物理服务器时作用相同。强制关闭可能不会始终关闭所有正在运行的进程，如果用这种方式关闭 VM，可能会有丢失数据的风险。
挂起	将正在运行的 VM 挂起在其当前状态，并将此状态保存在默认存储库中的某个文件里。此选项可让您关闭 VM 的主机服务器，在重新启动后恢复 VM，从而将其还原到原始运行状态。
恢复	恢复挂起的 VM 并还原其原始运行状态。
开始	在 VM 关闭后启动（也称为冷启动）。

如果电源控制操作失败，请将鼠标悬停在警报上，此时将显示一条弹出消息，其中包含有关故障的详细信息。

在相应的管理员执行映像维护任务时，使用维护模式临时阻止新连接。

在计算机上启用维护模式后，将不允许新连接，直到禁用该模式。如果用户已登录，维护模式将在所有用户注销后生效。对于未注销的用户，请发送一条消息，通知他们计算机将在某个特定时间关闭，并使用电源控制项强制关闭计算机。

1. 从用户详细信息视图选择计算机，或在过滤器视图选择一组计算机。
2. 选择维护模式，然后打开选项。

如果用户尝试连接到分配的桌面但此桌面处于维护模式，将显示一条消息，指示此桌面当前不可用。无法进行新连接，直到您禁用维护模式。

对用户问题进行故障排除

May 04, 2016

使用 Director 的“技术支持”视图（[活动管理器](#)页面）查看用户相关信息：

- 检查与用户登录、连接和应用程序相关的详细信息。
- 重影用户计算机。
- 执行下表中建议的操作对问题进行故障排除，并将其上报给相应的管理员。

故障排除提示

用户的问题	相应的建议：
登录时间过长，或者间歇或重复性地出现登录失败。	诊断用户登录问题
应用程序运行缓慢或不响应	解决应用程序故障
连接失败	还原桌面连接
会话执行缓慢或不响应	还原会话
视频加载缓慢或画质差	运行 HDX 通道系统报告

注意：为确保计算机不处于维护模式，请从用户详细信息视图查看计算机详细信息面板。

搜索提示

当您在“搜索”字段中键入用户名时，Director 会在 Active Directory 中跨所有配置为支持 Director 的站点搜索用户。

在“搜索”字段中输入多用户计算机名称时，Director 显示特定计算机的计算机详细信息。

在“搜索”字段中输入端点名称时，Director 使用连接到指定端点的未经身份验证（匿名）的会话和经过身份验证的会话，从而对未经身份验证的会话进行故障排除。请确保端点名称唯一以启用对未经身份验证的会话进行故障排除。

搜索结果中也包括当前未使用计算机或未分配给计算机的用户。

- 搜索时不区分大小写。
- 不完整的输入会产生一个可能匹配的列表。
- 您键入一个由两部分构成且中间以空格分隔的名称（用户名、姓名或显示名称）的几个字母后，搜索结果中将包含与这两个字符串均匹配的条目。例如，如果您键入jo rob，搜索结果中可能包括“John Robertson”或 Robert, Jones 等字符串。

要返回登录页面，请单击 Director 徽标。

将故障诊断信息上载给 Citrix 技术支持部门

从单个 Delivery Controller 或 VDA 运行 Citrix Scout 可捕获关键数据点和 Citrix Diagnosis Facility (CDF) 跟踪以对所选计算机进行故障排除。捕获此信息后，Scout 会安全地将数据点上载给 Citrix 技术支持部门。工具即服务 (Tools As a Service, TaaS) 平台可以利用此信息缩短客户所报告问题的解决时间。

Scout 随 XenApp 或 XenDesktop 组件一起安装。安装或升级到 XenDesktop 7.1、XenDesktop 7.5 或 XenApp 7.5 时，Scout 出现在 Windows 的“开始”菜单或者 Windows 8 或 8.1 的“开始”屏幕中。

要启动 Scout，请从“开始”菜单或“开始”屏幕中选择 Citrix > Citrix Scout。

有关使用和配置 Scout 的信息以及常见问题解答，请参阅 <http://support.citrix.com/article/CTX130147>。

以下视频概述了如何使用 Scout。

重影用户

May 04, 2016

在 Director 中，使用重影用户功能直接在用户虚拟机或会话中进行查看和操作。用户必须连接到您要执行重影操作的计算机。可通过检查用户标题栏中所列的计算机名称来验证此项操作。

1. 在用户详细信息视图中，选择用户会话。
2. 为所选用户会话激活重影：
 - 要执行计算机监视，请在活动管理器视图中，单击重影。
 - 要执行会话监控，请在用户详细信息视图中，查找会话详细信息面板，然后单击重影。
3. 连接初始化后，将显示一个对话框，提示您打开或保存.msrcincident文件。
4. 请使用远程协助查看器（如果默认情况下尚未选择）打开事件文件。此时将在用户设备上显示一个确认提示窗口。
5. 指导用户单击是启动计算机或会话共享。

要执行其他控制，请要求用户共享键盘和鼠标控制。

将 Microsoft Internet Explorer 浏览器配置为：通过远程协助客户端自动打开已下载的 Microsoft 远程协助 (.msra) 文件。

为此，您必须在组策略编辑器中启用文件下载自动提示设置：

计算机配置 > 管理模板 > Windows 组件 > Internet Explorer > Internet 控制面板 > 安全页面 > Internet 区域 > 文件下载自动提示。

默认情况下，此选项对本地 Intranet 区域中的站点启用。如果 Director 站点不在本地 Intranet 区域中，请考虑将该站点手动添加到此区域。

向用户发送消息

May 04, 2016

在 Director 中，向连接到一台或多台计算机的用户发送消息。例如，使用此功能发送有关管理操作（如即将发生的桌面维护、计算机注销和重新启动以及配置文件重置）的即时通知。

1. 在活动管理器视图中，选择用户，然后单击详细信息。
2. 在用户详细信息视图中，查找会话详细信息面板，然后单击发送消息。
3. 在主题和消息字段中键入您的消息信息，然后单击发送。

如果消息发送成功，将在 Director 中显示确认消息。如果用户计算机已连接，则将在其中显示消息。

如果消息未发送成功，则将在 Director 中显示错误消息。根据错误消息对问题进行故障排除。完成后，再次键入主题和消息文本，然后单击重试。

诊断用户登录问题

May 04, 2016

请按照以下通用步骤进行操作：

1. 从用户详细信息视图，使用登录时长面板对登录状态进行故障排除。
 - 如果用户正在登录，视图会反映登录进度。
 - 如果用户当前已登录，登录时长面板会显示用户登录当前会话所用的时间。
2. 让用户注销然后重新登录，以便您检查登录时长数据。此面板通常在大约 3 分钟后更新，这一时间可能会更长，具体取决于完成登录所需的时间。
3. 检查登录过程的各个阶段：
 - **代理** — 决定向用户分配哪个桌面所用的时间。
 - **VM 启动** — 启动桌面所用的时间。
 - **HDX 连接** — 建立 HDX 连接所用的时间（具体取决于网络情况）。
 - **GPO** — 应用组策略对象所用的时间。
 - **登录脚本** — 脚本所用的时间。
 - **配置文件加载** — 加载用户配置文件所用的时间。
 - **交互式会话** — 建立交互式用户会话所用的时间。

总登录时间并不是这些阶段的精确总和。例如，一些阶段并行发生，而在某些阶段中会发生额外处理，这可能会导致登录时长大于阶段总和。

提示：要在图表中找到异常值或意外值，请将当前会话每个阶段所用的时间与最近七天此用户的平均时长以及最近七天此交付组所有用户的平均时长进行比较。

如有必要请进行上报。例如，如果 VM 启动速度缓慢，问题可能存在于虚拟机管理程序中，因此您可以将问题上报给虚拟机管理程序管理员。而如果代理速度缓慢，您可以将问题上报给站点管理员，让其检查 Delivery Controller 上的负载平衡情况。

故障排除提示：检查异常差异，其中包括：

- 缺少（当前）登录栏
- 当前时长与此用户平均时长之间存在很大差异。原因可能包括：
 - 安装了新的应用程序。
 - 发生了操作系统更新。
 - 更改了配置。
- 用户的登录次数（当前时长和平均时长）与交付组平均时长之间存在很大差异。

如果需要，请单击重新启动，观察用户的登录过程，以对问题进行故障排除，比如 VM 启动或代理方面的问题。

解决应用程序故障

May 04, 2016

在活动管理器视图中，单击应用程序选项卡。您可以查看此用户访问的所有计算机上的所有应用程序，其中包括当前连接的计算机的本地应用程序和托管应用程序，以及每个应用程序的当前状态。

注意：如果应用程序选项卡处于灰显状态，请联系有权启用此选项卡的管理员。

列表仅包含已经在会话中启动的应用程序。

对于服务器操作系统计算机和桌面操作系统计算机，会列出各个已断开连接的会话的应用程序。如果用户未建立连接，将不会显示任何应用程序。

操作	说明
终止不响应的应用程序。	选中不响应的应用程序并单击结束应用程序。终止应用程序后，请求用户重新启动应用程序。
终止不响应的进程。	如果您拥有必需的权限，请单击进程选项卡。选择与应用程序相关的进程或者占用大量 CPU 资源或内存的进程，然后单击结束进程。 但是，如果您没有终止进程所需的权限，尝试结束进程操作将失败。
重新启动用户的计算机。	对于所选会话，请单击重新启动，此操作仅适用于桌面操作系统计算机。 或者，从计算机详细信息视图，使用电源控制项重新启动或关闭计算机。指导用户再次登录，以便您重新检查应用程序。 对于服务器操作系统计算机，重新启动选项不可用，而是需要用户注销，然后再重新登录。
将计算机置于维护模式。	如果计算机的映像需要维护（如添加补丁或进行其他更新），请将计算机置于维护模式并将问题上报给相应的管理员。从计算机详细信息视图中，单击详细信息，然后启用维护模式选项。上报给相应的管理员。

还原桌面连接

May 04, 2016

从 Director 的用户标题栏检查当前计算机的用户连接状态。

如果桌面连接失败，将会显示导致连接失败的错误，以帮助您确定如何进行故障排除。

操作	说明
请确保计算机未处于维护模式。	请确保在用户详细信息页面上已关闭维护模式。
重新启动用户的计算机。	选择计算机，然后单击重新启动。如果用户计算机没有响应或无法连接，比如计算机占用异常高的 CPU 资源（这会导致 CPU 不可用），请使用此选项。

还原会话

May 04, 2016

如果会话断开连接，它将继续处于活动状态，其应用程序仍会运行，但用户设备将不再与该服务器通信。

在用户详细信息视图的会话详细信息面板中，对会话故障进行故障排除。您可以查看当前会话的详细信息（以会话 ID 指示）。

操作	说明
终止不响应的应用程序或进程。	单击应用程序选项卡。选中不响应的应用程序并单击结束应用程序。同样，选中对应的不响应的进程并单击结束进程。 此外，结束占用过多内存或 CPU 资源的进程，因为这种进程可能会导致 CPU 无法使用。
断开 Windows 会话的连接。	单击会话控制并选择断开连接。此选项仅适用于代理服务器操作系统计算机。对于非代理会话，则禁用此选项。
从会话中注销用户。	单击会话控制并选择注销。

要测试会话，用户可尝试再次登录会话。您也可以重影该用户，更密切地监控此会话。

注意：如果用户设备正在运行的 Virtual Delivery Agent (VDA) 版本早于 XenDesktop 7，Director 将无法显示有关该会话的完整信息；相反，将会显示一条关于无法获取信息的信息。这些消息可能会在用户详细信息页面和活动管理器中显示。

运行 HDX 通道系统报告

May 04, 2016

在用户详细信息视图的 HDX 面板中，检查用户计算机上 HDX 通道的状态。只有在用户计算机使用 HDX 连接时，才可使用此面板。

如果出现了一条指示当前无法获取信息的信息，请等待一分钟以便页面进行刷新，或选择刷新按钮。HDX 数据更新时间比其他数据更新时间稍长。

单击错误或警告图标，以了解更多信息。

提示：可以在同一对话框中，通过单击标题栏左角的向左和向右箭头，查看其他通道的相关信息。

HDX 通道系统报告主要供 Citrix 技术支持用来进行进一步的故障排除。

1. 在 HDX 面板中，单击下载系统报告。
2. 可以查看或保存 .xml 报告文件。
 - 要查看 .xml 文件，请单击打开。 .xml 文件将出现在 Director 应用程序所在的窗口中。
 - 要保存 .xml 文件，请单击保存。此时将显示另存为窗口，提示您提供 Director 计算机上的文件下载位置。

重置 Personal vDisk

May 04, 2016

警告：重置磁盘时，各设置会还原为出厂默认值，并且磁盘中的所有数据都将删除，包括应用程序。除非修改了 Personal vDisk 默认设置（用于重定向来自 C: 驱动器的配置文件），或不是使用第三方配置文件解决方案，否则将保留此配置文件数据。

要进行重置，必须运行启用 Personal vDisk 的计算机；但用户无需登录该计算机。

此选项仅适用于桌面操作系统计算机；对服务器操作系统计算机则禁用此选项。

1. 从技术支持视图中，选择目标桌面操作系统计算机。
2. 从此视图中或在用户详细信息视图的个性化面板中，单击重置 Personal vDisk。
3. 单击重置。此时将显示一条警告消息，提醒用户将注销。用户注销（如果用户已登录）之后，计算机将重新启动。

如果重置成功，用户详细信息视图个性化面板中的 Personal vDisk 状态字段值将为正在运行。如果重置失败，将在正在运行值的右侧显示一个红色的 X。指向此 X 时，将显示失败的相关信息。

重置用户配置文件

May 04, 2016

警告：重置配置文件时，虽然用户的文件夹和文件都已保存并复制到新的配置文件，但大部分用户配置文件数据仍将被删除（例如，注册表被重置，应用程序设置可能被删除）。

1. 从 Director，搜索要重置其配置文件的用户，并选择此用户的会话。
2. 单击**重置配置文件**。
3. 指示用户从所有会话中注销。
4. 指导用户重新登录。从用户配置文件保存的文件夹和文件已复制到新的配置文件。

重要：如果用户在多个平台（如 Windows 8 和 Windows 7）上具有配置文件，请指导用户首先重新登录用户报告有问题的同一桌面或应用程序。这样可确保重置正确的配置文件。

如果此配置文件是 Citrix 用户配置文件，那么它在用户桌面显示时已重置。如果此配置文件是 Microsoft 漫游配置文件，文件夹还原可能短时间内仍在进行。在还原完成前，用户必须保持登录状态。

注意：上述步骤假定您使用的是 XenDesktop（桌面 VDA）。如果您使用的是 XenApp（服务器 VDA），则需要登录才能执行配置文件重置。用户随后需要注销，然后重新登录才能完成配置文件重置。

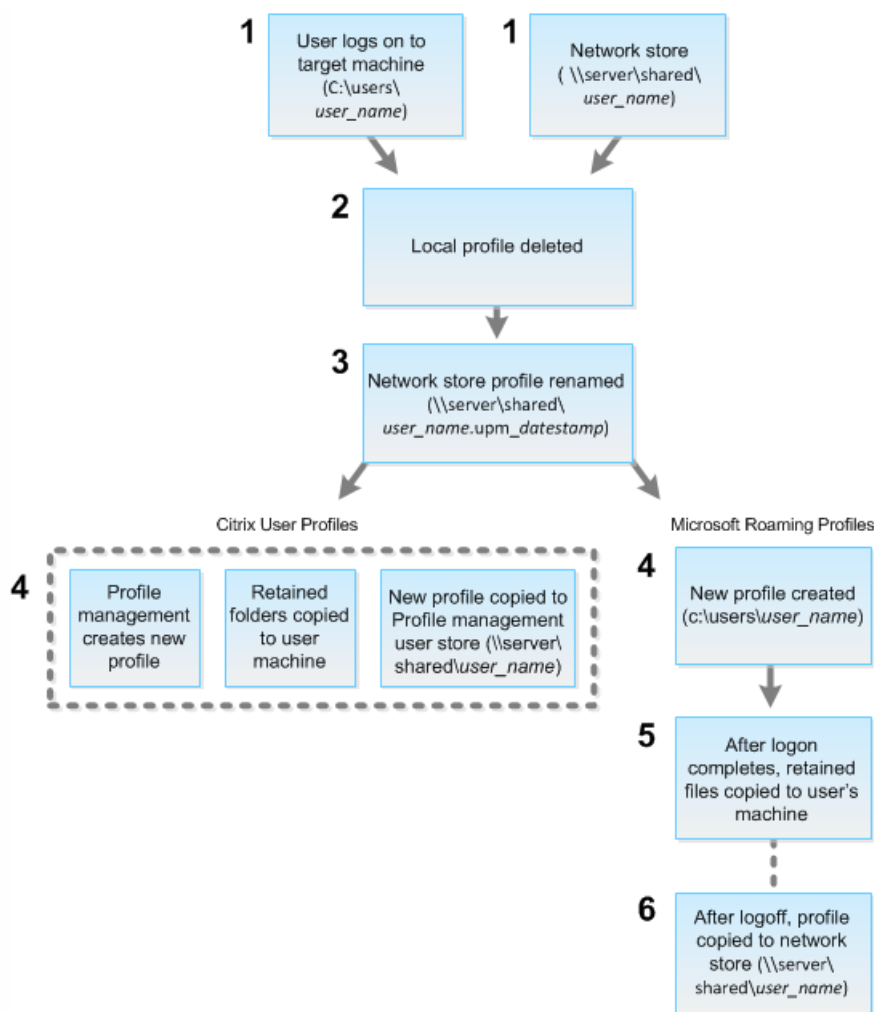
如果配置文件未能成功重置（例如，用户无法成功重新登录计算机或部分文件已丢失），您必须手动还原原始配置文件。用户配置文件中的文件夹（及其文件）将保存并复制到新配置文件中。将按照所列顺序复制这些文件：

- 桌面
- Cookies
- 收藏夹
- 文档
- 图片
- 音乐
- 视频

注意：在 Windows 8 或更高版本中，重置配置文件时不会复制 Cookie。

所有 Citrix 用户配置文件或 Microsoft 漫游配置文件均可重置。在用户注销并且您选择重置命令（在 Director 中或使用 PowerShell SDK）后，Director 首先识别正在使用的用户配置文件并发出相应的重置命令。Director 通过 Profile Management 接收信息，包括有关配置文件大小、类型和登录时间的信息。

用户下次登录时，此图会显示发生的处理过程。



1. Director 发出的重置命令会指定配置文件类型。然后，Profile Management Service 将尝试重置此类型的配置文件，并查找相应的网络共享（用户存储）。如果用户由 Profile Management 处理，但却接收到漫游配置文件命令，用户将被拒绝（反之亦然）。
2. 如果存在本地配置文件，则会将其删除。
3. 重命名网络配置文件。
4. 下一步操作取决于要重置的配置文件是 Citrix 用户配置文件还是 Microsoft 漫游配置文件。
 - 对于 Citrix 用户配置文件，将使用 Profile Management 导入规则创建新配置文件，然后将文件夹复制回网络配置文件，之后用户可以继续正常登录。如果将漫游配置文件用于重置，则漫游配置文件中的任何注册表设置将保留在重置配置文件中。
注意：如果需要，您可以配置 Profile Management，以使模板配置文件覆盖漫游配置文件。
 - 对于 Microsoft 漫游配置文件，使用 Windows 创建新配置文件，然后在用户登录时，将文件夹复制回用户设备。用户再次注销时，新配置文件将复制到网络存储中。

1. 指示用户从所有会话中注销。
2. 删除本地配置文件（如果存在）。
3. 查找网络共享上的存档文件夹，即文件夹名称中包含日期和时间且扩展名为 .upm_datestamp 的文件夹。
4. 删除当前配置文件名称，即不包含 upm_datestamp 扩展名的文件。
5. 使用原配置文件名称重命名存档文件夹，即删除日期和时间扩展名。此时已将配置文件恢复为其重置之前的原始状态。

Session Recording - 适用于 XenApp 7.6 FP1、FP2 和 LTSR

May 04, 2016

通过 Session Recording，您可以在遵守公司政策和法规的前提下，通过任意类型的连接录制服务器操作系统 VDA 计算机托管的任意用户会话的屏幕活动。Session Recording 对会话进行记录、编录和存档，以便进行检索和播放。

Session Recording 使用灵活的策略自动触发应用程序会话的录制。这样就可以使 IT 能够监视和检查应用程序（如金融运营和医疗患者信息系统）的用户活动，在遵守法规和安全监视方面加强内部控制。同样，Session Recording 也通过加快问题识别及问题解决时间在技术支持方面予以帮助。

通过记录和监视增强安全性。Session Recording 允许组织录制处理敏感信息的应用程序屏幕上的用户活动。这对于医疗和金融等管制行业来说尤为重要。其中包括禁止录制的个人信息，策略控制允许进行选择性录制。

强大的活动监视。Session Recording 可以捕获屏幕更新，并对这些更新进行存档（包括鼠标活动，以及安全视频录制中可见的击键输出），从而记录特定用户、应用程序及服务器的活动。

Session Recording 并非为收集法律诉讼证据而设计的。Citrix 建议使用 Session Recording 的组织使用其他技术进行证据收集，例如传统的视频录制，结合传统的以文字为基础的 eDiscovery 工具。

更快地解决问题。用户由于难以重现问题而致电技术支持时，客服人员可启用用户会话记录。再次出现该问题时，Session Recording 会提供该错误的时间戳直观记录，该记录可用于更加快速地进行故障排除。

Session Recording 入门

May 04, 2016

执行以下步骤之后，即可开始录制和查看 XenApp 会话。

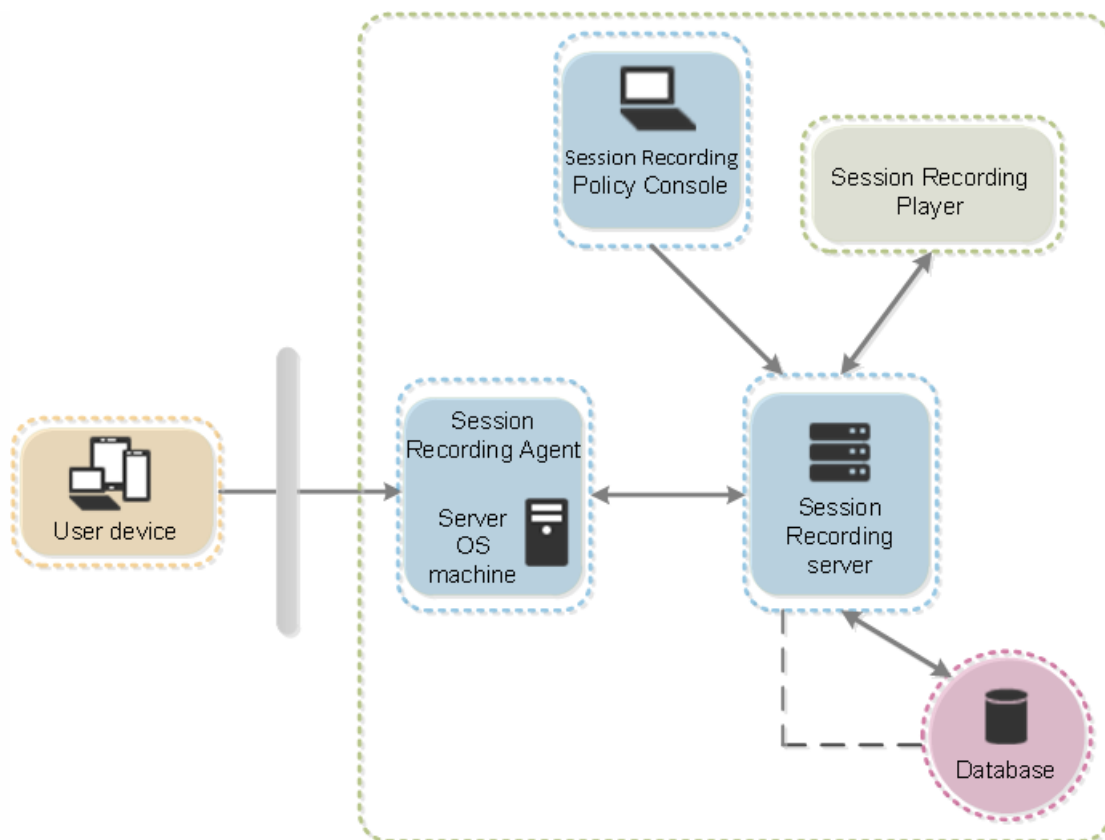
1. 熟悉 Session Recording 组件。
2. 选择适用于您的环境的部署方案。
3. 验证安装要求。
4. 安装 Session Recording。
5. 配置 Session Recording 组件以允许录制和查看会话。

Session Recording 包括 5 个组件：

- **Session Recording Agent**。安装在每台服务器操作系统计算机上的组件，用于启用录制。负责录制会话数据。
- **Session Recording Server**。用于托管以下各项的服务器：
 - **Broker**。IIS 6.0+ 托管的 Web 应用程序，可处理 Session Recording Player 发出的搜索查询和文件下载请求，处理 Session Recording 策略控制台发出的策略管理请求，以及评估每个 XenApp 会话的录制策略。
 - **存储管理器**。管理从每个已启用 Session Recording 且运行 XenApp 的计算机接收到的录制会话文件的 Windows 服务。
- **Session Recording Player**。一种用户界面，用户可从工作站进行访问以播放录制的 XenApp 会话文件。
- **Session Recording 数据库**。用于存储录制的会话数据的 SQL 数据库。
- **Session Recording 策略控制台**。用于创建策略以指定要录制的会话的控制台。

下图显示了 Session Recording 的各个组件以及各组件之间的关系：

在此处显示的部署示例中，Session Recording Agent、Session Recording Server、Session Recording 数据库、Session Recording 策略控制台以及 Session Recording Player 均位于安全防火墙后面。Session Recording Agent 安装在服务器操作系统计算机上。第二台服务器用于托管 Session Recording 策略控制台，第三台服务器用作 Session Recording Server，第四台服务器用于托管 Session Recording 数据库。Session Recording Player 安装在工作站上。位于防火墙外的客户端设备与安装了 Session Recording Agent 的服务器操作系统计算机进行通信。在防火墙内部，Session Recording Agent、Session Recording 策略控制台、Session Recording Player 以及 Session Recording 数据库都与 Session Recording Server 进行通信。



计划部署

May 04, 2016

可以通过不同的方案对 Session Recording 组件进行部署，具体取决于您的环境。

Session Recording 部署不必限制为单个站点。除 Session Recording Agent 外，其他所有组件均独立于服务器站点。例如，您可以配置多个站点使用单个 Session Recording Server。

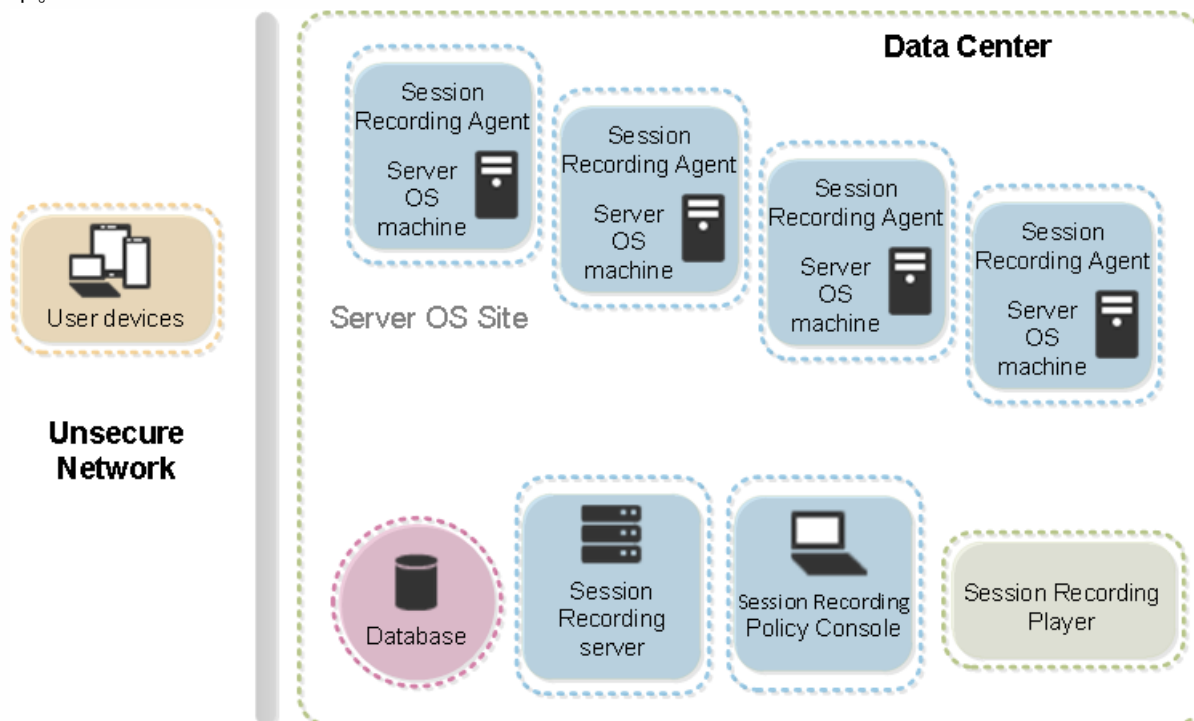
或者，如果您拥有包含多个代理的大型站点，并计划录制许多图形密集型应用程序（例如，AutoCAD 应用程序），或者您有许多会话要录制，则可能对 Session Recording Server 具有较高性能要求。要缓解性能问题，可以在不同的计算机上安装多个 Session Recording Server，并将 Session Recording Agent 指向这些不同的计算机。请记住，一次只能将一个代理指向一台服务器。

以下是推荐的两种 Session Recording 部署配置：

- 在单台服务器操作系统计算机上部署 Session Recording Agent。
- 在站点上的多台服务器操作系统计算机上部署 Session Recording Agent。

服务器站点部署

此类部署用于录制一个或多个站点的会话。在站点中每台服务器操作系统计算机上安装 Session Recording Agent。站点位于安全防火墙后面的数据中心。Session Recording Administration 组件（Session Recording 数据库、Session Recording Server、Session Recording 策略控制台）安装在其他服务器上，Session Recording Player 安装在工作站上，全部位于防火墙背后，而非位于数据中心中。XenApp 客户端（例如工作站、移动设备以及便携式计算机）位于防火墙之外的不安全网络环境中。



安全性建议

Jan 04, 2017

Session Recording 可以部署在一个安全网络中，并由管理员进行访问，因此，Session Recording 非常安全。即开即用部署设计为简单易用，且具有可选择性配置的数字签名和加密等安全功能。

Session Recording 组件之间的通信通过 Internet Information Services (IIS) 以及 Microsoft 消息队列 (MSMQ) 实现。IIS 在每个 Session Recording 组件之间提供网络服务通信链接。MSMQ 提供了可靠的数据传输机制，可用于从 Session Recording Agent 向 Session Recording Server 发送录制的会话数据。

规划部署时，请考虑如下安全性建议：

- 确保运行 Session Recording 组件的服务器在物理上是安全的。如有可能，请将这些计算机锁在一个安全的房间内，只有授权人员能够直接取用。
- 隔离在单独子网或域上运行 Session Recording 组件的服务器。
- 在 Session Recording Server 与其他服务器之间安装防火墙，防止访问其他服务器的用户访问录制的会话数据。
- 安装 Microsoft 提供的最新安全更新，保持 Session Recording Admin Server 和 SQL 数据库为最新版本。
- 限制非管理员登录到管理计算机。
- 严格限制授权人员更改录制策略及查看录制的会话。
- 安装数字证书，使用 Session Recording 文件签名功能，并在 IIS 中设置 SSL 通信。
- 通过将 Session Recording Agent Properties 对话框中列出的 MSMQ 协议设置为 HTTPS，设置 MSMQ 使用 HTTPS 作为传输协议。有关详细信息，请参阅 [MSMQ 故障排除](#)。
- 在 Session Recording Server 和 Session Recording 数据库上使用 TLS 1.0 并禁用 SSLv2、SSLv3 和 RC4 密码。有关详细信息，请参阅 Microsoft 文章 <http://support.microsoft.com/default.aspx?scid=kb;en-us;187498> 和 <http://support.microsoft.com/kb/245030/zh-cn>。
- 使用播放保护功能。播放保护是一项 Session Recording 功能，在文件下载到 Session Recording Player 前对文件进行加密。默认情况下，此选项处于启用状态，且位于“Session Recording Server Properties”中。
- 按照加密密钥长度和加密算法的 NSIT 指南进行操作。

有关配置 Session Recording 功能的信息，请参阅 <http://support.citrix.com/article/CTX200868>。

-

在安装了 Session Recording Server 的计算机上，IIS Web 服务器在从 Session Recording Agent、Session Recording Player 或 Session Recording 策略控制台建立 SSL 连接时，将其服务器证书发送至客户端。收到服务器证书后，Session Recording Agent、Session Recording Player 或策略控制台会确定颁发该证书的证书颁发机构 (CA) 以及客户端是否信任该 CA。如果客户端不信任该 CA，证书将被拒绝，并且将在 Session Recording Agent 的应用程序事件日志中记录错误，或者在 Session Recording Player 或策略控制台中向用户显示一条错误消息。

服务器证书是通过收集服务器相关信息并请求 CA 为该服务器颁发证书来安装的。请求服务器证书时必须指定正确的信息，并确保正确指定服务器的名称。如果使用完全限定的域名 (FQDN) 连接客户端 (Session Recording Agent、Session Recording Player 以及策略控制台)，则指定给 CA 的证书信息必须使用服务器的 FQDN 而非 NetBIOS 名称。如果指定 NetBIOS 名称，请勿在请求服务器证书时指定 FQDN。将服务器证书安装到本地服务器的证书存储中。在每个连接的客户端上安装颁发的 CA 证书。

您的组织可能拥有私有 CA 来颁发可与 Session Recording 一起使用的服务器证书。如果您使用私有 CA，请确保每台客户端设备上均安装有颁发的 CA 证书。有关使用证书以及证书颁发机构的信息，请参阅 Microsoft 文档。或者，某些公司和组织目前

担当 CA 的角色，包括 VeriSign、Baltimore、Entrust 及其各自的附属机构。

所有证书都拥有一个由 CA 定义的过期日期。要查找该过期日期，请查看证书的属性。请确保在过期日期之前续订证书，以防止 Session Recording 中产生任何错误。

默认情况下，Session Recording 安装会配置为使用 HTTPS，您需要使用 CA 颁发的服务器证书来配置默认 Web 站点。如果需要有关在 IIS 中安装服务器证书的说明，请查阅 IIS 文档。

可扩展性注意事项

May 04, 2016

除运行 XenApp 所需的资源外，安装和运行 Session Recording 几乎不需要其他资源。但是，如果您计划使用 Session Recording 录制大量会话，或者如果您计划录制的会话会导致出现大量会话文件（例如，图形密集型应用程序），请在规划 Session Recording 部署时注意系统的性能。

有关构建高度可扩展的 Session Recording 系统的详细信息，请参阅 <http://support.citrix.com/article/CTX200869>。

在本文中：

硬件建议

磁盘和存储硬件

网络容量

计算机处理能力

部署多台 Session Recording Server

数据库可扩展性

考虑要向每台 Session Recording Server 发送的数据量，以及服务器可以处理和存储此数据的速率。系统可以存储输入数据的速率必须快于数据输入速率。

要估算数据输入速率，请将录制的会话数乘以每个录制的会话的平均大小，然后除以录制会话所需的时间。例如，在 8 小时工作日中，您可能录制了 5000 个 Microsoft Outlook 会话，每个会话的大小为 20MB。在这种情况下，数据输入速率约为 3.5MBps。（5000 个会话乘以 20MB，除以 8 小时，再除以每小时 3600 秒。）

通过优化单个 Session Recording Server 的性能或在不同计算机上安装多个 Session Recording Server，可以提升性能。

磁盘和存储硬件是规划 Session Recording 部署时要考虑的最重要的因素。存储解决方案的写入性能尤为重要。数据写入磁盘的速度越快，系统的整体性能越高。

适合使用 Session Recording 的存储解决方案包括由本地磁盘控制器或连接的存储区域网络 (SAN) 作为 RAID 阵列控制的一组本地磁盘。

注意：由于存在与向网络驱动器写入录制数据有关的性能和安全问题，因此，请勿将 Session Recording 与网络连接存储 (NAS) 一起使用。

对于本地驱动器设置，具有内置缓存内存的磁盘控制器的性能会有所增强。缓存磁盘控制器必须具有电池备份装置，以在电源出现故障时确保数据完整性。

100Mbps 的网络链接适合连接 Session Recording Server。千兆位以太网连接可能会提高性能，但不会将性能比 100Mbps 链接提高 10 倍。

确保 Session Recording 不与可能会争夺可用网络带宽的第三方应用程序共享网络交换机。理想情况下，网络交换机专用于

Session Recording Server。

对于安装了 Session Recording Server 的计算机，请考虑以下规格：

- 建议使用双 CPU 或双核 CPU
- 建议使用 2GB 到 4GB RAM

超过这些规格不会显著提高性能。

如果单台 Session Recording Server 不满足性能要求，您可以在不同计算机上安装多台 Session Recording Server。在此类型的部署中，每台 Session Recording Server 都具有其自己的专用存储、网络交换机以及数据库。要分发负载，在部署中将 Session Recording Agent 指向不同的 Session Recording Server。

Session Recording 数据库需要安装 Microsoft SQL Server 2014、Microsoft SQL Server 2012 或 Microsoft SQL Server 2008 R2。发送到该数据库的数据量非常少，因为该数据库仅存储与录制的会话有关的元数据。录制的会话本身的文件会写入到单独的磁盘中。通常情况下，每个录制的会话只需约 1KB 数据库空间，除非使用 Session Recording 事件 API 将可搜索事件插入到会话中。

Express Edition 版本的 Microsoft SQL Server 2014、Microsoft SQL Server 2012 和 Microsoft SQL Server 2008 R2 将数据库大小限制为 10GB。以每个录制会话大小为 1KB 计算，该数据库可编录大约 400 万个会话。Microsoft SQL Server 的其他版本没有数据库大小限制，仅由可用磁盘空间限制。数据库中会话数量逐渐增多，但数据库的性能以及搜索速度的降低几乎可以忽略不计。

如果没有通过 Session Recording 事件 API 进行自定义，每个录制的会话将生成四个数据库事务：录制启动时生成两个、用户登录到正在录制的会话时生成一个、录制结束时生成一个。如果使用 Session Recording 事件 API 对会话进行了自定义，录制的每个可搜索事件都会生成一个事务。由于即使是最基本的数据库部署在一秒钟内也可处理数百个交易，因此数据库上的处理负载不可能非常重。影响非常低，Session Recording 数据库可以与其他数据库（包括 XenApp 或 XenDesktop 数据存储数据库）在相同的 SQL Server 上运行。

如果 Session Recording 部署需要在数据库中编录数百万个录制的会话，请按照 Microsoft 对 SQL Server 可扩展性的指南进行操作。

重要部署注意事项

May 04, 2016

- 为使 Session Recording 组件能够互相通信，请确保将这些组件安装在同一个域中，或者能够相互传递信任关系的不同可信域中。不能将系统安装在工作组中，也不能跨具有外部信任关系的域进行安装。
- Session Recording 不支持在一次部署中使用两个或更多 Session Recording Server 组成群集。
- 由于播放大型录制件时图形比较密集，并且内存使用率高，Citrix 建议不要将 Session Recording Player 作为已发布应用程序进行安装。
- 为 Session Recording 安装配置 SSL/HTTPS 通信。确保在 Session Recording Server 上安装证书，并且根证书颁发机构 (CA) 在 Session Recording 组件上受信任。
- 如果在运行 SQL Server 2014 Express Edition 或 SQL Server 2012 Express Edition 或 SQL Server 2008 R2 Express Edition 的独立服务器上安装 Session Recording 数据库，该服务器必须启用 TCP/IP 协议，且必须运行 SQL Server Browser 服务。默认情况下，这些设置处于禁用状态，但必须启用它们才能使 Session Recording Server 与数据库进行通信。有关启用这些设置的信息，请参阅 Microsoft 文档。
- 规划 Session Recording 部署时，请考虑会话共享的效果。已发布应用程序的会话共享可能会与已发布应用程序的 Session Recording 录制策略规则相冲突。Session Recording 将活动策略与用户打开的第一个已发布应用程序相匹配。用户打开第一个应用程序之后，随后在同一会话中打开的任何应用程序都将继续遵循对第一个应用程序有效的策略。例如，如果某策略指出只应录制 Microsoft Outlook，则录制将在用户打开 Outlook 时开始。不过，如果用户接着打开已发布的 Microsoft Word（在 Outlook 运行时），也会对 Word 进行录制。相反，如果活动策略没有指定应该对 Word 进行录制，并且用户在启动 Outlook（根据策略应进行录制）之前启动 Word，则不会对 Outlook 进行录制。

安装 Session Recording

May 04, 2016

本文在单独的章节中介绍 XenApp 7.6 Feature Pack 2 和 LTSR 以及 Feature Pack 1 的安装说明。

预安装核对表

开始安装之前，请确保完成以下列表中的步骤：

	步骤
	开始安装之前，请安装必备项。请参阅 系统要求 。
	选择要安装每个 Session Recording 组件的计算机，并确保每台计算机都满足要安装的组件对硬件和软件的要求。
	从 LTSR 下载页面中下载 Session Recording zip 文件。
	如果使用 SSL 协议在 Session Recording 组件之间进行通信，请在您的环境中安装正确的证书。请参阅 安装证书 。
	为 Session Recording 组件安装任何所需的修补程序。可以从 Citrix 支持 页面下载修补程序。
	配置 Director 以创建并激活 Session Recording 策略。

注意：

- Citrix 建议您根据录制策略将已发布的应用程序分到独立的交付组中，因为如果已发布的应用程序位于相同的交付组中，其会话共享可能会与活动策略冲突。Session Recording 将活动策略与用户打开的第一个已发布应用程序相匹配。
- 如果要将 Machine Creation Services (MCS) 或 Provisioning Services 与 XenApp 结合使用，请为服务器准备一个唯一的 QMID。请参阅“已知问题”中的说明。执行此步骤失败可能会导致录制数据丢失。
- SQL Server 要求启用 TCP/IP、运行 SQL Server Browser 服务并使用 Windows 身份验证。
- 如果要使用 HTTPS，请为 SSL/HTTPS 配置服务器证书。

Session Recording 安装文件

需要从 Citrix 下载页面下载以下安装文件：

- Session Recording Administration 文件
 - Broker_PowerShellSnapIn_x64.msi
 - SessionRecordingAdministrationx64.msi
- Session Recording Agent 文件
 - SessionRecordingAgentx64.msi

- Session Recording Player 文件
 - SessionRecordingPlayer.msi

Session Recording Administration 组件包括 Session Recording 数据库、Session Recording Server 和 Session Recording 策略控制台。可以选择要在服务器上安装上述哪些组件。

开始安装 Session Recording Administration 组件之前，请务必安装所有必备项。请参阅 [Session Recording Administration 组件](#)。

要提高安全性，可以在安装数据库后删除这些权限。

1. 运行 **Broker_PowerShellSnapIn_x64.msi** 并按照说明完成安装。
2. 以管理员身份启动 Windows 命令提示窗口，然后运行该命令：

```
msiexec /i SessionRecordingAdministrationx64.msi
```

或双击 .msi 文件。

3. 在安装用户界面上，选择“下一步”并接受许可协议。
4. 在“Session Recording Administration 安装程序”屏幕上，选择要安装的 Session Recording Administration 组件。

开始安装 Session Recording 数据库之前，请务必安装所有必备项。请参阅 [Session Recording Administration 组件](#)。

1. 在“数据库配置”页面上：
 - 如果要在同一台服务器上安装所有 Administration 组件，在“Session Recording Server 名称”字段中键入 **localhost**。
 - 如果要在不同的服务器上安装 Session Recording Server 和 Session Recording 数据库，用以下格式键入托管 Session Recording Server 的计算机的名称：域\计算机名称。Session Recording Server 名称为用于访问数据库的用户帐户。

Citrix Session Recording Administration Setup

Database and Server Configuration

Specify the instance name of the Session Recording Database and the computer account of the Session Recording Server.

Database Instance Name

Test

Example: .\SQLEXPRESS, machine-name\SQLEXPRESS, machine-name

Session Recording Server computer account

Example: localhost, domain\machine-name

< Back Next > Cancel

如果数据库实例不是您在设置实例时配置的已命名实例，则只能使用 SQL Server 的计算机名称。如果您已为实例命名，请使

用计算机名称\实例名称作为数据库实例名称。要确定使用的服务器实例名称，请在 SQL Server 上运行 `select @@servername`，返回值为准确的数据库实例名称。

单击**测试**，测试与 SQL Server 的连接。确保当前用户具有公共 SQL Server 角色权限；否则，测试将由于权限限制而失败。然后，单击**下一步**继续安装。

2. 按照说明完成安装。安装期间，如果当前用户不是数据库管理员，则会显示一个对话框，表示需要数据库管理员的凭据和 `sysadmin` 服务器权限。请输入正确的凭据，然后单击**确定**以继续安装。安装将创建新 Session Recording 数据库，并添加 Session Recording Server 的计算机帐户 `db-owner`。

开始安装 Session Recording Server 之前，请务必安装所有必备项。请参阅 [Session Recording Administration 组件](#)。

1. 在“数据库实例名称”文本框中输入 SQL Server 的名称。如果要使用命名实例，请输入计算机名称\实例名称，否则，请仅输入计算机名称。
2. 单击“测试”，测试与 SQL Server 的连接。确保当前用户具有公共 SQL Server 角色权限；否则，测试将由于权限限制而失败。然后，单击**下一步**以继续安装，并按照说明完成安装。
3. 在安装向导的结尾，您可以选择是否参与 Citrix 客户体验改善计划。加入此计划时，匿名统计信息和使用情况信息将发送到 Citrix；有关详细信息，请参阅[关于 Citrix 客户体验改善计划 \(CEIP\)](#)。

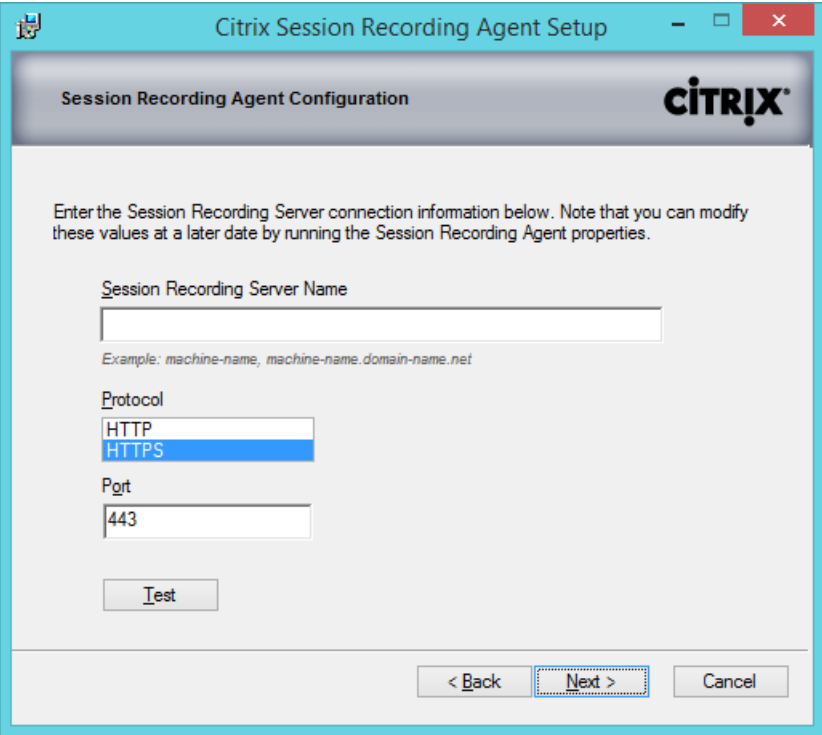
必须在要录制会话的服务器操作系统 VDA 计算机上安装 Session Recording Agent。

1. 使用 Server Manager 安装 .NET Framework 3.5 和 Microsoft Message Queuing (MSMQ)，并在 XenApp 7.6 服务器操作系统 VDA 上启用 HTTP 支持。
2. 以管理员身份启动 Windows 命令提示窗口，然后运行该命令：

```
msiexec /i SessionRecordingAgentx64.msi
```

或双击 .msi 文件。

3. 在安装用户界面上，选择**下一步**并接受许可协议。
4. 在“Session Recording Agent 配置”页面上，输入安装了 Session Recording Server 的计算机名称，以及用于连接到 Session Recording Server 的协议和端口信息。



Session Recording 默认安装使用 HTTPS/SSL 来确保通信安全。如果未配置 SSL，请使用 HTTP。为此，请导航到 Session Recording Broker 站点，在 IIS 管理控制台中取消选中 SSL。打开 SSL 设置，取消选中“需要 SSL”复选框。

5. 对于 Feature Pack 2：如果要使用回滚功能，请安装更新 ICATS760WX64010 (<http://support.citrix.com/article/CTX142037>)。对于 LTSR：回滚功能包含在 VDA (7.6.300) 的 LTSR 版本中。因此，如果您的 VDA 为 LTSR 版本，则可以使用该功能，无需进一步进行操作。

6. 按照说明完成安装。

在 Session Recording Server 或域中的一个或多个工作站上为查看会话录制件的用户安装 Session Recording Player。

- 运行 SessionRecordingPlayer.msi 并按照说明完成安装。

要从 XenApp 7.6 Feature Pack 1 或 2 升级 Session Recording，请在安装了相应 Session Recording 功能的计算机上运行安装程序。

按照说明完成升级安装。升级过程中无需提供任何信息。

预安装核对表

开始安装之前，请确保完成以下列表中的步骤：

	步骤

	开始安装之前，请安装必备项。请参阅系统要求。 步骤 选择要安装每个 Session Recording 组件的计算机，并确保每台计算机都满足要安装的组件对硬件和软件的要求。
	从 XenApp > 组件下的 Citrix 下载页面下载 Session Recording zip 文件。
	如果使用 SSL 协议在 Session Recording 组件之间进行通信，请在您的环境中安装正确的证书。请参阅安装证书。
	为 Session Recording 组件安装任何所需的修补程序。可以从 Citrix 支持 页面下载修补程序。
	配置 Director 以创建并激活 Session Recording 策略。

注意：

- Citrix 建议您根据录制策略将已发布的应用程序分到独立的交付组中，因为如果已发布的应用程序位于相同的交付组中，其会话共享可能会与活动策略冲突。Session Recording 将活动策略与用户打开的第一个已发布应用程序相匹配。
- 如果要将 Machine Creation Services (MCS) 或 Provisioning Services 与 XenApp 结合使用，请为服务器准备一个唯一的 QMId。请参阅已知问题中的说明。执行此步骤失败可能会导致录制数据丢失。
- SQL Server 要求启用 TCP/IP、运行 SQL Server Browser 服务并使用 Windows 身份验证。
- 如果要使用 HTTPS，请为 SSL/HTTPS 配置服务器证书。

Session Recording 安装文件

需要从 Citrix 下载页面下载以下安装文件：

- Session Recording Administration 文件
 - Broker_PowerShellSnapIn_x64.msi
 - SessionRecordingAdministrationx64.msi
- Session Recording Agent 文件
 - SessionRecordingAgentx64.msi
- Session Recording Player 文件
 - SessionRecordingPlayer.msi

Session Recording Administration 组件包括 Session Recording 数据库、Session Recording Server 和 Session Recording 策略控制台。可以选择要在服务器上安装上述哪些组件。

开始安装 Session Recording Administration 组件之前，请务必安装所有必备项。请参阅 [Session Recording Administration 组件](#)。

在本地 SQL Server 上安装 Session Recording 数据库时，您必须使用具有 sysadmin SQL Server 角色权限的 NT AUTHORITY\SYSTEM 登录。

在远程 SQL Server 上安装 Session Recording 数据库时，您必须使用具有 sysadmin SQL Server 角色权限的运行安装程序的计算机帐户登录。

要提高安全性，可以在安装数据库后删除这些权限。

1. 运行 Broker_PowerShellSnapIn_x64.msi 并按照说明完成安装。
2. 以管理员身份启动 Windows 命令提示窗口，然后运行该命令：

```
msiexec /i SessionRecordingAdministrationx64.msi
```

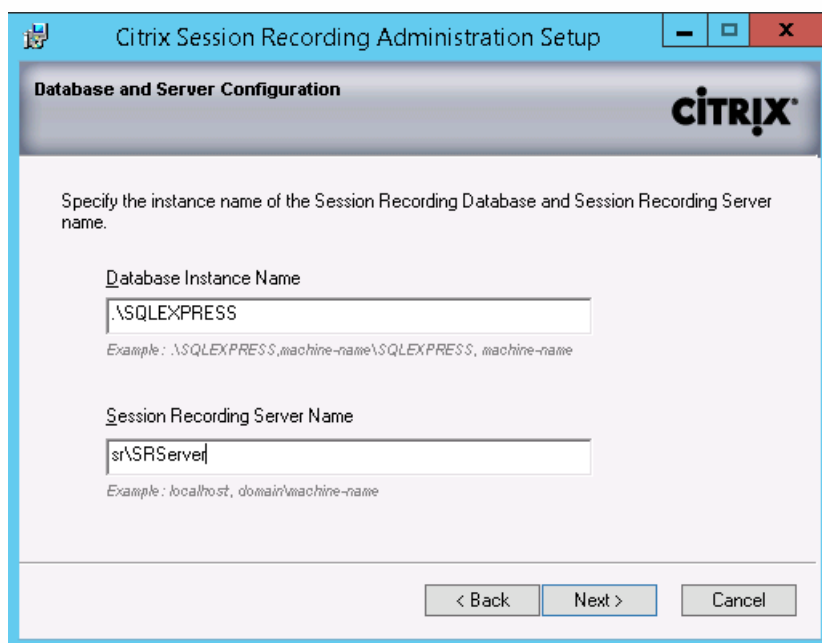
或双击 .msi 文件。

3. 在安装用户界面上，选择下一步并接受许可协议。
4. 在“Session Recording Administration 安装程序”屏幕上，选择要安装的 Session Recording Administration 组件。

开始安装 Session Recording 数据库之前，请务必安装所有必备项。请参阅 [Session Recording Administration 组件](#)。

1. 在“数据库配置”页面上：

- 如果要在同一台服务器上安装所有 Administration 组件，在“Session Recording Server 名称”字段中键入 **localhost**。
- 如果要在不同的服务器上安装 Session Recording Server 和 Session Recording 数据库，用以下格式键入托管 Session Recording Server 的计算机的名称：**域\计算机名称**。Session Recording Server 名称为用于访问数据库的用户帐户。



如果数据库实例不是您在设置实例时配置的已命名实例，则只能使用 SQL Server 的计算机名称。如果您已为实例命名，请使用 **计算机名称\实例名称** 作为数据库实例名称。要确定使用的服务器实例名称，请在 SQL Server 上运行 **select @@servername**，返回值为准确的数据库实例名称。

2. 按照说明完成安装。安装将创建新 Session Recording 数据库，并添加 Session Recording Server 的计算机帐户 **db-owner**。

安装 Session Recording Server

开始安装 Session Recording Server 之前，请务必安装所有必备项。请参阅 [Session Recording Administration 组件](#)。

- 在数据库实例名称文本框中输入 SQL Server 的名称。如果要使用命名实例，请输入计算机名称\实例名称，否则，请仅输入

计算机名称。

必须在要录制会话的服务器操作系统 VDA 计算机上安装 Session Recording Agent。

1. 使用 Server Manager 安装 .NET Framework 3.5 和 Microsoft Message Queuing (MSMQ)，并在 XenApp 7.6 服务器操作系统 VDA 上启用 HTTP 支持。
2. 以管理员身份启动 Windows 命令提示窗口，然后运行该命令：

```
msiexec /i SessionRecordingAgentx64.msi
```

或双击 .msi 文件。

3. 在安装用户界面上，选择“下一步”并接受许可协议。
4. 在 Session Recording Agent 配置页面上，输入安装了 Session Recording Server 的计算机名称，以及用于连接到 Session Recording Server 的协议和端口信息。



Session Recording 默认安装使用 HTTPS/SSL 来确保通信安全。如果未配置 SSL，请使用 HTTP。为此，请导航到 Session Recording Broker 站点，在 IIS 管理控制台中取消选中 SSL。打开 SSL 设置，取消选中需要 SSL 复选框。

5. 如果要使用滚动功能，请务必安装以下修补程序：<http://support.citrix.com/article/CTX142037>。

6. 按照说明完成安装。

在 Session Recording Server 或域中的一个或多个工作站上为查看会话录制件的用户安装 Session Recording Player。

- 运行 SessionRecordingPlayer.msi 并按照说明完成安装。

要从服务器或工作站中删除 Session Recording 组件，请使用 Windows“控制面板”中的卸载或删除程序功能。要删除 Session Recording 数据库，您必须具有与安装时同样的 sysadmin SQL Server 角色权限。

可以使用 Director 控制台创建并激活 Session Recording 策略。

1. 对于 https 连接，请在 Director 服务器的可信根证书中安装证书以信任 Session Recording Server。
2. 要将 Director 服务器配置为使用 Session Recording Server，请运行以下命令：
C:\inetpub\wwwroot\Director\tools\DirectorConfig.exe /configsessionrecording
3. 在 Director 服务器上输入 Session Recording Server 的 IP/FQDN、从 Session Recording Agent 到 Session Recording Broker 的端口和连接类型 (http/https)。

自动安装

May 04, 2016

要在多台服务器上安装 Session Recording Agent，请编写一个进行无提示安装脚本。

以下命令行可安装 Session Recording Agent，并创建一个日志文件，用于捕获安装信息。

`msiexec /i SessionRecordingAgentx64.msi sessionrecordingservername=yourservername sessionrecordingbrokerprotocol=yourbrokerprotocol sessionrecordingbrokerport=yourbrokerport /!v yourinstalla`
其中：

`yourservername` 是指 NetBIOS 名称或托管 Session Recording Server 的计算机的 FQDN。如未指定，则此值默认为 `localhost`。

`yourbrokerprotocol` 是指 HTTP 或 HTTPS，表示 Session Recording Agent 用于与 Session Recording Broker 进行通信的协议，如未指定，则此值默认为 HTTPS。

`yourbrokerport` 是一个整数，表示 Session Recording Agent 用于与 Session Recording Broker 进行通信的端口。如未指定，则此值默认为零，从而指示 Session Recording Agent 使用所选协议的默认端口号：对于 HTTP 使用 80，对于 HTTPS 使用 443。

`/!v`指定详细模式日志记录

`yourinstallationlog` 是指创建安装日志文件的位置。

`/q`指定无提示模式。

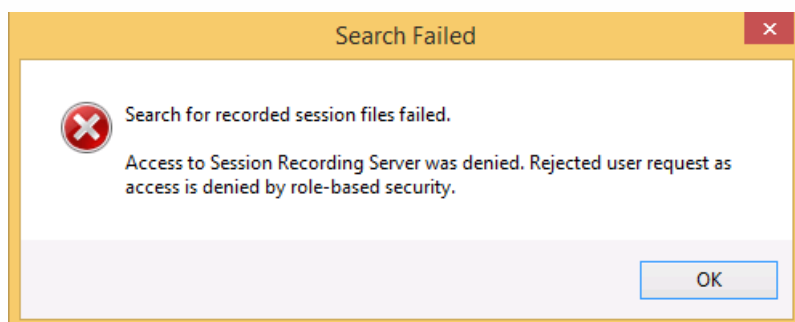
配置用于播放和录制会话的 Session Recording

May 04, 2016

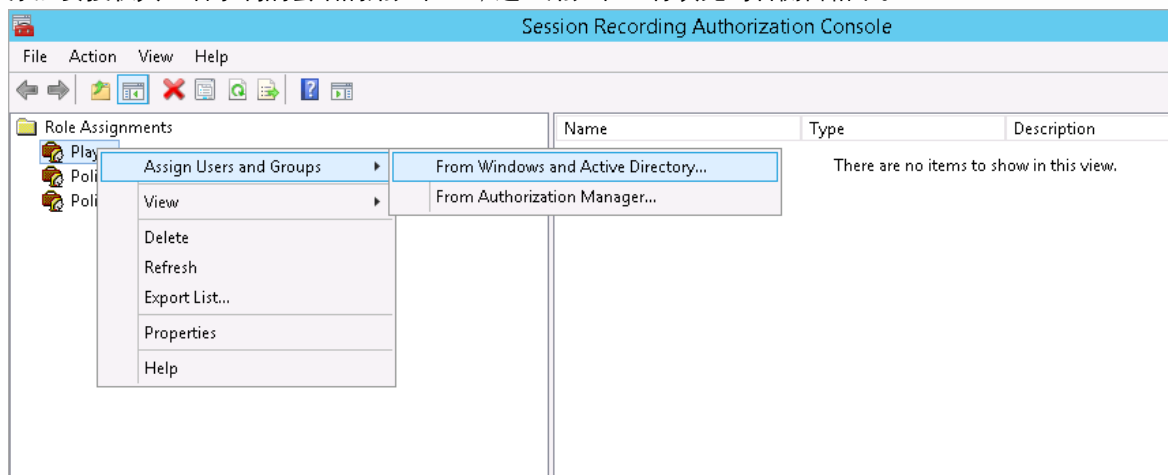
安装 Session Recording 组件后，执行以下步骤以将 Session Recording 配置为录制 XenApp 会话并允许用户查看这些会话：

- 授权用户播放录制件
- 授权用户管理录制策略
- 将活动录制策略更改为录制会话的策略
- 配置用于连接至 Session Recording Server 的 Session Recording Player

安装 Session Recording 时，所有用户都没有播放录制的会话权限。必须向包括管理员在内的每位用户分配权限。如果用户没有播放录制的会话所需的权限，则在尝试播放录制的会话时会收到以下错误消息：



1. 以管理员身份登录托管 Session Recording Server 的计算机。
2. 启动 Session Recording Authorization 控制台。
3. 在 Session Recording Authorization 控制台上，选择播放器。
4. 添加要授权其查看录制的会话的用户和组，这些用户和组将填充到右侧窗格中。



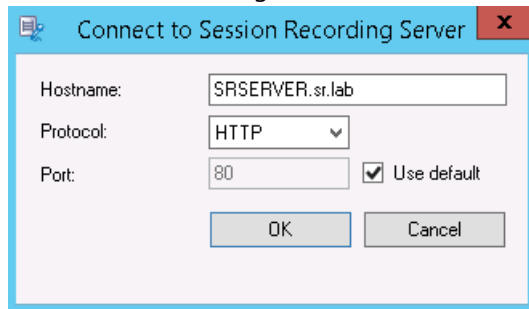
安装 Session Recording 时，域管理员默认授予控制录制策略的权限。您可以更改授权设置。

1. 以管理员身份登录托管 Session Recording Server 的计算机。
2. 启动 Session Recording Authorization 控制台并选择 策略管理员。
3. 添加能够管理录制策略的用户和组。

活动的录制策略指定在安装了 Session Recording Agent 并连接到 Session Recording Server 的所有服务器操作系统 VDA 上的会话录制行为。安装 Session Recording 时，活动录制策略为不录制。更改活动录制策略之后才会录制会话。

1. 以授权策略管理员身份登录安装了 Session Recording 策略控制台的服务器。
2. 启动 Session Recording 策略控制台。
3. 如果系统显示连接到 Session Recording Server 弹出窗口提示，请确保托管 Session Recording Server 的计算机名称、协议和

端口正确无误。



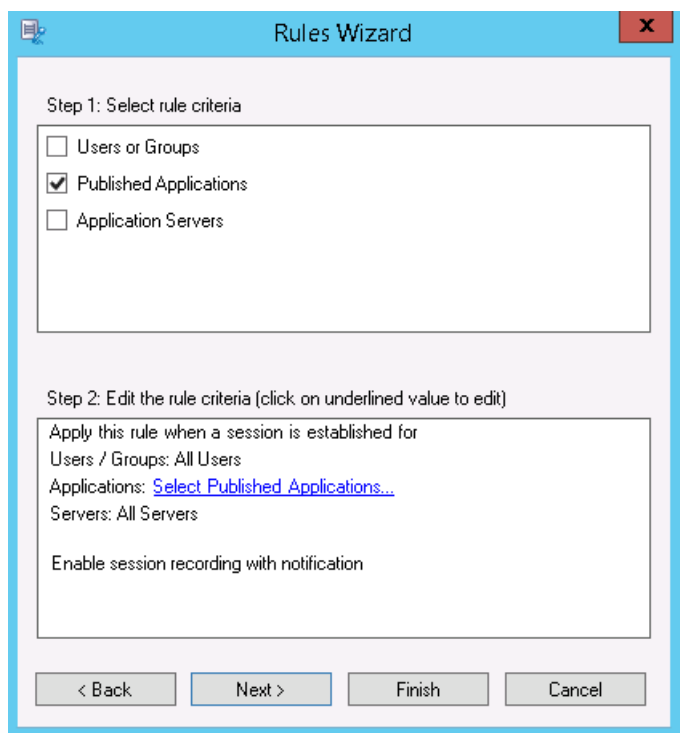
4. 在 Session Recording 策略控制台中，展开录制策略。此时将显示安装 Session Recording 时提供的录制策略，以及一个复选标记，用于指示处于活动状态的策略：
 - 不录制。此为默认策略。如果未指定其他策略，则不会录制会话。
 - 录制每个人并通知。如果选择此策略，则会录制所有会话。系统将显示一个弹出窗口，通知用户正在录制其会话。
 - 录制每个人但不通知。如果选择此策略，则会录制所有会话。系统不显示用于通知用户正在录制其会话的弹出窗口。
5. 选择要指定为活动策略的策略。
6. 从菜单栏中，依次选择操作 > 激活策略。

注意：Session Recording 允许您创建自己的录制策略。创建录制策略时，创建的策略将在 Session Recording 策略控制台内的“录制策略”文件夹中显示。

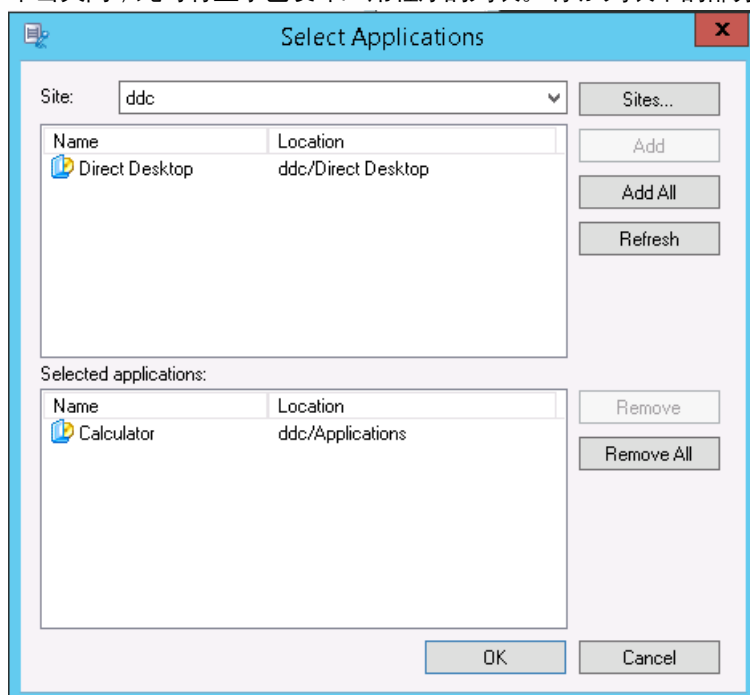
通用录制策略可能不满足您的要求。可以根据用户、VDA 服务器和应用程序来配置策略。

重要：一个策略中可以包含多条规则，但一次只能运行一条活动策略。

1. 以授权策略管理员身份登录安装了 Session Recording 策略控制台的服务器。
2. 启动 Session Recording 策略控制台，然后选择录制策略 > 添加新策略。
3. 右键单击新建策略，然后选择添加新规则。
4. 在“规则”向导中，选择启用会话录制并通知，然后单击下一步。
5. 选中复选框已发布的应用程序，然后单击选择已发布的应用程序超链接。



6. 在“选择应用程序”屏幕上，单击站点和添加。
7. 输入 XenApp 7.6 FP1 交付控制器的名称。
8. 单击连接，此时站点将进行枚举。
9. 单击关闭，此时将显示已发布应用程序的列表。添加列表中的部分应用程序，然后单击确定和完成。



10. 在策略上单击鼠标右键，然后选择激活。如有需要，可以重命名策略。

在 Session Recording Player 可以播放会话之前，必须对其进行配置，使其连接到存储已录制会话的 Session Recording Server。每个 Session Recording Player 均可配置为能够与多台 Session Recording Server 连接，但一次只能连接到一个 Session Recording Server。如果将 Player 配置为能够与多台 Session Recording Server 连接，用户可以通过选中复选框更改

Player 连接到的 Session Recording Server。

1. 登录到安装了 Session Recording Player 的工作站。
2. 启动 Session Recording Player。
3. 从 Session Recording Player 菜单栏，选择工具 > 选项。
4. 在连接选项卡中，单击添加。
5. 在主机名字段中，键入托管 Session Recording Server 的计算机名称或 Internet 协议 (IP) 地址，然后选择协议。默认情况下，Session Recording 配置为使用 HTTPS/SSL 来确保通信安全。如果未配置 SSL，请选择“HTTP”。
6. 如果要配置 Session Recording Player，使其能够连接到多台 Session Recording Server，请对每台 Session Recording Server 重复步骤 4 和 5。
7. 请务必选中要连接到的 Session Recording Server 对应的复选框。

授予用户访问权限

May 04, 2016

注意：出于安全原因，请只授予用户执行特定功能（例如查看录制的会话）所需的权限。

可以通过使用 Session Recording Server 上的 Session Recording Authorization 控制台为 Session Recording 用户分配角色来授予其访问权限。Session Recording 用户具有三个角色：

- **播放者**。具有查看录制的 XenApp 会话的权限。此角色中没有默认成员身份。
- **策略查询者**。允许托管 Session Recording Agent 的服务器请求录制策略评估。默认情况下，经过身份验证的用户属于此角色的成员。
- **策略管理员**。具有查看、创建、编辑、删除以及启用录制策略的权限。默认情况下，托管 Session Recording Server 的计算机管理员属于此角色的成员。

Session Recording 支持在 Active Directory 中定义的用户和组。

1. 以管理员或策略管理员角色的成员身份登录到托管 Session Recording Server 的计算机。
2. 启动 Session Recording Authorization 控制台。
3. 选择要分配给用户的角色。
4. 依次选择操作 > Assign Windows Users and Groups（分配 Windows 用户和组）。
5. 添加用户和组。

对控制台所做的任何更改都将于每分钟执行一次的更新期间生效。

创建并激活录制策略

May 04, 2016

使用 Session Recording 策略控制台可创建并激活用于确定要录制的会话的策略。

可以在安装 Session Recording 时激活可用的系统策略，也可以创建并激活自己的自定义策略。Session Recording 系统策略将单个规则应用于所有用户、已发布应用程序以及服务器。自定义策略指定要录制的用户、已发布应用程序以及服务器。

活动策略确定要录制的会话。每次只能有一个策略处于活动状态。

Session Recording 提供了以下系统策略：

- **不录制。**如果选择此策略，则不会录制会话。此为默认策略；如果未指定其他策略，则不录制会话。
- **录制每个人并通知。**如果选择此策略，则会录制所有会话。系统将显示一个弹出窗口，通知用户正在录制其会话。
- **录制每个人但不通知。**如果选择此策略，则会录制所有会话。系统不显示用于通知用户正在录制其会话的弹出窗口。

无法修改或删除系统策略。

1. 登录到安装了 Session Recording 策略控制台的服务器。
2. 启动 Session Recording 策略控制台。
3. 如果出现连接到 Session Recording Server 弹出窗口提示，请确保 Session Recording Server 名称、协议和端口正确无误。单击确定。
4. 在 Session Recording 策略控制台中，展开录制策略。
5. 选择要指定为活动策略的策略。
6. 从菜单栏中，依次选择操作 > 激活策略。

创建您自己的策略时，可以制定规则以指定录制哪些用户和组、已发布应用程序以及服务器的会话。Session Recording 策略控制台中的向导可帮助您创建规则。要获取已发布的应用程序和服务器列表，您必须具有站点管理员读取权限。在此站点的 Delivery Controller 上配置该权限。

对于创建的每条规则，您都需要指定录制操作以及规则条件。录制操作将应用到满足规则条件的会话。

对于每条规则，请选择一种录制操作：

- **不录制。**（在规则向导中选择禁用会话记录。）此记录操作将指定不记录满足规则条件的会话。
- **录制并通知。**（在规则向导中选择启用会话记录并通知。）此记录操作将指定记录满足规则条件的会话。系统将显示一个弹出窗口，通知用户正在录制其会话。
- **录制但不通知。**（在规则向导中选择启用会话记录而不通知。）此记录操作将指定记录满足规则条件的会话。用户不知道系统正在录制其会话。

对于每条规则，请至少选择以下选项之一来创建规则条件：

- **用户或组。**创建要应用规则的录制操作的用户或组列表。
- **已发布应用程序。**创建要应用规则的录制操作的已发布应用程序列表。在规则向导中，选择可提供这些应用程序的一个或多个 XenApp 站点。

- 应用程序服务器。您可以创建一个服务器操作系统计算机列表来应用该规则中的录制操作。在规则向导中，选择服务器所在的一个或多个 XenApp 站点。

在一条录制策略中创建多条规则时，某些会话可能满足多条规则的条件。在这些情况下，具有最高优先级的规则会应用到该会话。

规则的录制操作决定其优先级：

- 不录制操作的规则优先级最高
- 录制并通知操作的规则优先级次之
- 录制但不通知操作的规则优先级最低

某些会话可能不满足录制策略中的任何规则条件。对于这些会话，将应用策略回退规则的录制操作。回退规则的录制操作始终为不录制。无法修改或删除回退规则。

使用 Active Directory 组

Session Recording 允许您在创建策略时使用 Active Directory 组。使用 Active Directory 组代替单个用户可简化规则和策略的创建和管理。例如，如果公司财务部门中的用户包含在名为“Finance”的 Active Directory 组中，那么您可以通过在创建规则时在规则向导中选择“Finance”组来创建此组的所有成员要应用的规则。

将用户列入白名单

可以创建 Session Recording 策略，确保绝不会录制组织中某些用户的会话。这称为将这些用户列入白名单。对于负责处理隐私相关信息的用户或当您所在的组织不希望录制某些级别的员工会话时，列入白名单功能很有用。

例如，如果公司内的所有经理都属于名为“Executive”的 Active Directory 组中的成员，那么您可以创建规则来禁用“Executive”组的会话录制，从而确保这些用户的会话绝不会被录制。包含此规则的策略处于活动状态时，不会录制“Executive”组成员的会话。组织内其他成员的会话根据活动策略中的其他规则进行录制。

创建新策略

注意：使用规则向导时，系统可能会在未显示有下划线的值时提示您“单击有下划线的值以编辑”。只有适用时才显示有下划线的值。如果未显示有下划线的值，请忽略此步骤。

1. 登录到安装了 Session Recording 策略控制台的服务器。
2. 启动 Session Recording 策略控制台。
3. 如果系统提示连接到 Session Recording Server 弹出窗口，请确保 Session Recording Server 名称、协议和端口正确无误。单击 OK（确定）。
4. 在 Session Recording 策略控制台中，选择录制策略。
5. 从菜单栏中，依次选择操作 > 添加新策略。左侧窗格中将显示名为新策略的策略。
6. 选择新策略，然后从菜单栏中依次选择操作 > 重命名。
7. 键入要创建的策略名称，然后按 Enter 或单击新名称之外的任意位置。
8. 选中该策略后，从菜单栏中依次选择操作 > 添加新规则以启动规则向导。
9. 按照说明为此策略创建规则。

修改策略

1. 登录到安装了 Session Recording 策略控制台的服务器。
2. 启动 Session Recording 策略控制台。
3. 如果系统提示连接到 Session Recording Server 弹出窗口，请确保 Session Recording Server 名称、协议和端口正确无误。单击 OK（确定）。

- 4. 在 Session Recording 策略控制台中，展开录制策略。
- 5. 选择要修改的策略。策略规则将显示在右侧窗格中。
- 6. 添加新规则、修改规则或删除规则：
 - 从菜单栏中，依次选择操作 > 添加新规则。如果策略处于活动状态，系统将显示一个弹出窗口，请求您确认该操作。请使用规则向导创建新规则。
 - 选择要修改的规则，单击鼠标右键，并选择属性。请使用规则向导修改规则。
 - 选择要删除的规则，单击鼠标右键，并选择删除规则。

删除策略

- 注意：无法删除系统策略或处于活动状态的策略。
- 1. 登录到安装了 Session Recording 策略控制台的服务器。
 - 2. 启动 Session Recording 策略控制台。
 - 3. 如果系统提示连接到 Session Recording Server 弹出窗口，请确保 Session Recording Server 名称、协议和端口正确无误。单击 OK（确定）。
 - 4. 在 Session Recording 策略控制台中，展开录制策略。
 - 5. 在左侧窗格中，选择要删除的策略。如果该策略处于活动状态，必须激活其他策略。
 - 6. 从菜单栏中，依次选择操作 > 删除策略。
 - 7. 选择是确认该操作。

激活策略之后，之前的活动策略仍然有效，直至用户的会话结束。但在某些情况下，文件滚动后新策略才生效。文件达到最大大小限制时会发生滚动。有关录制的最大文件大小的信息，请参阅[指定录制的文件大小](#)。

下表详细说明了在录制会话并发生滚动的过程中应用新策略时会出现的情况：

如果之前的策略为：	新策略为：	滚动后策略将变为：
不录制	任何其他策略	不更改。仅当用户登录到新会话时新策略才生效。
录制但不通知	不录制	停止录制。
	录制并通知	继续录制并显示一条通知消息。
录制并通知	不录制	停止录制。
	录制但不通知	继续录制。下次用户登录时不显示任何消息。

禁用或启用录制

May 04, 2016

在要录制会话的每台服务器操作系统计算机上安装 Session Recording Agent。可通过每个代理中的设置对安装了该代理的服务器启用录制。启用录制之后，Session Recording 会评估活动录制策略，从而确定要录制的会话。

安装 Session Recording Agent 时，将启用录制。Citrix 建议您禁用未录制的服务器上的 Session Recording，因为即使未进行录制，这些服务器的性能也会受到些许影响。

1. 登录到安装了 Session Recording Agent 的服务器。
2. 从“开始”菜单中，选择 Session Recording Agent Properties。
3. 在会话录制下，选中或清除对此服务器操作系统 VDA 启用会话录制复选框，以指定是否录制此服务器的会话。
4. 系统提示时，重新启动 Session Recording Agent 服务以接受所做的更改。

注意：安装 Session Recording 时，活动策略为不录制（任何服务器上都不录制会话）。要开始录制，请使用 Session Recording 策略控制台激活其他策略。

配置与 Session Recording Server 的连接

May 04, 2016

安装 Session Recording Agent 时，通常会配置 Session Recording Agent 与 Session Recording Server 之间的连接。要在安装 Session Recording Agent 后配置此连接，请使用 Session Recording Agent Properties。

1. 登录到安装了 Session Recording Agent 的服务器。
2. 从开始菜单中，选择 Session Recording Agent Properties。
3. 单击连接选项卡。
4. 在 Session Recording Server 字段中，键入服务器名称或其 Internet 协议 (IP) 地址。
5. 在 Session Recording Storage Manager 消息队列部分，选择 Session Recording Storage Manager 用于进行通信的协议并修改默认端口号（如有必要）。
6. 在消息生存期字段中，接受默认值 7200 秒（2 小时），或键入一个新值，表示每条消息在队列中保留的秒数（如果通信失败）。超过此时间段之后，将删除该消息，且文件可播放，直至数据丢失点停止。
7. 在 Session Recording Broker 部分，选择 Session Recording Broker 用来通信的通信协议，并修改默认端口号（如果需要）。
8. 系统提示时，重新启动 Session Recording Agent 服务接受所做的更改。

创建通知消息

May 04, 2016

如果活动录制策略指定录制用户的会话时应向用户发送通知，则系统会在用户键入其凭据后弹出一个窗口，显示通知消息。默认通知消息如下：“正在录制您在最近启动的一个或更多程序中执行的活动。如果您不同意此条件，请关闭这些程序。”用户单击确定可关闭该窗口并继续进行会话。

默认通知消息会以托管 Session Recording Server 的计算机的操作系统使用的语言显示。

可以您所选的语言创建自定义通知，但一种语言只能创建一条通知消息。通知消息所用的语言与用户的首选区域设置相对应。

1. 登录到托管 Session Recording Server 的计算机。
2. 从开始菜单中，选择 Session Recording Server Properties。
3. 在 Session Recording Server Properties 中，单击通知选项卡。
4. 单击 添加。
5. 选择消息使用的语言并键入新消息。一种语言只能创建一条消息。

接受并激活后，新消息将显示在特定于语言的通知消息框中。

启用自定义事件录制

May 04, 2016

Session Recording 允许您使用第三方应用程序将自定义数据（称为事件）插入到已录制的会话中。使用 Session Recording Player 查看会话时会显示这些事件。它们属于已录制会话文件的一部分，无法在录制会话之后对其进行修改。

例如，某个事件可能包含以下文本：“用户打开浏览器”。用户每次在录制会话过程中打开浏览器时，都会将该文本插入到录制件中。使用 Session Recording Player 播放该会话时，查看器可以通过记录 Session Recording Player 的事件和书签列表中显示的标记数，来查找并计算用户打开某个浏览器的次数。

将自定义事件插入到服务器上的录制件中：

- 可使用“Session Recording Agent Properties”启用要插入自定义事件的每台服务器上的设置。必须分别启用每台服务器；不能全局启用站点中的所有服务器。
- 写入在每个用户的 XenApp 会话内部运行的事件 API 上构建的应用程序（以将数据插入到录制件中）。

Session Recording 安装包括一个事件录制 COM 应用程序 (API)，允许您将文本从第三方应用程序插入到录制件中。可以使用以多种编程语言（包括 Visual Basic、C++ 或 C#）编写的 API。Session Recording Event API .dll 作为 Session Recording 安装的一部分进行安装。可以在 C:\Program Files\Citrix\SessionRecording\Agent\Bin\Interop.UserApi.dll 下找到该文件。

启用服务器上的自定义事件记录

1. 登录到安装了 Session Recording Agent 的服务器。
2. 从开始菜单中，选择 Session Recording Agent Properties。
3. 在 Session Recording Agent Properties 中，单击录制选项卡。
4. 在自定义事件录制下，选择允许第三方应用程序将自定义数据录制到此 XenApp 服务器上复选框。

启用或禁用实时会话播放

May 04, 2016

使用 Session Recording Player 可以在录制会话之后或录制会话过程中对其进行查看。查看当前正在录制的会话与查看实时进行的操作类似；但是，从 XenApp 服务器传送数据的过程中实际上存在 1 至 2 秒的延迟。

查看未完成录制的会话时某些功能不可用：

- 完成录制之后才能分配数字签名。如果数字签名处于启用状态，则可以查看实时播放会话，但未对这些会话进行数字签名，并且完成录制之后您才能查看证书。
- 完成录制之后才能应用播放保护。如果播放保护处于启用状态，则可以查看实时播放会话，但完成会话之后您才能对其进行加密。
- 完成录制之后才能缓存文件。

默认情况下，实时会话播放处于启用状态。

1. 登录到托管 Session Recording Server 的计算机。
2. 从开始菜单中，选择 Session Recording Server Properties。
3. 在 Session Recording Server Properties 中，单击播放选项卡。
4. 选中或清除允许实时会话播放复选框。

启用或禁用播放保护

May 04, 2016

为了安全起见，Session Recording 将在 Session Recording Player 中下载并查看录制的文件之前自动对文件进行加密。此播放保护可防止未下载录制文件的任何用户复制和查看这些文件。无法在其他工作站上播放录制的文件，也无法由其他用户播放。加密的文件通过 .icle 扩展名进行标识，未加密的文件通过 .icl 扩展名进行标识。录制的文件在安装了 Session Recording Player 的工作站上的缓存中始终处于加密状态，直至授权用户将其打开。

Citrix 建议使用 HTTPS 来保护数据传输。

默认情况下，播放保护处于启用状态。

1. 登录到托管 Session Recording Server 的计算机。
2. 从开始菜单中，选择 Session Recording Server Properties。
3. 在 Session Recording Server Properties 中，单击播放选项卡。
4. 选中或清除加密下载供播放的会话录制文件复选框。

启用和禁用数字签名

May 04, 2016

如果您在安装了 Session Recording 组件的计算机上安装了证书，则可以通过向会话录制分配数字签名来增强 Session Recording 部署的安全性。

默认情况下，数字签名处于禁用状态。

启用数字签名

1. 登录到托管 Session Recording Server 的计算机。
2. 从开始菜单中，选择 Session Recording Server Properties。
3. 在 Session Recording Server Properties 中，单击签名选项卡。
4. 浏览至可以在安装了 Session Recording 组件的计算机之间实现安全通信的证书。

禁用数字签名

1. 登录到托管 Session Recording Server 的计算机。
2. 从开始菜单中，选择 Session Recording Server Properties。
3. 在 Session Recording Server Properties 中，单击签名选项卡。
4. 单击清除。

指定录制件的存储位置

May 04, 2016

使用“Session Recording Server Properties”可指定录制件的存储位置以及存档录制件的还原位置。

注意：要存档文件或还原删除的文件，请使用 icldb 命令。

指定记录文件的位置

默认情况下，录制件存储在托管 Session Recording Server 的计算机的驱动器:\SessionRecordings 目录中。可以更改录制件的存储目录，添加其他目录以平衡多个卷之间的负载，或者利用其他空间。列表中有多个目录表明录制件在多个目录之间处于负载平衡状态。可以多次添加一个目录。负载平衡在这些目录之间进行循环。

1. 登录到托管 Session Recording Server 的计算机。
2. 从开始菜单中，选择 Session Recording Server Properties。
3. 在 Session Recording Server Properties 中，单击存储选项卡。
4. 使用文件存储目录列表，来管理录制的存储目录。

可在本地驱动器上、SAN 卷上或 UNC 网络路径指定的位置处创建文件存储目录。网络映射的驱动器盘符不受支持。由于存在与向网络驱动器写入录制数据有关的严重性能和安全问题，因此，请勿将 Session Recording 与网络连接存储 (NAS) 一起使用。

指定存档文件的还原目录

默认情况下，存档的录制件还原到托管 Session Recording Server 的计算机的驱动器:\SessionRecordingsRestore 目录中。可以更改存档录制件的还原目录。

1. 登录到托管 Session Recording Server 的计算机。
2. 从开始菜单中，选择 Session Recording Server Properties。
3. 在 Session Recording Server Properties 中，单击存储选项卡。
4. 在归档文件的还原目录字段中，键入已还原的归档文件的目录。

指定录制的文件大小

May 04, 2016

录制文件大小越大，下载文件所需的时间越长，且在播放过程中使用搜寻滑块进行导航时文件的反应速度越慢。要控制文件的大小，请为文件指定阈值限制。录制达到此限制之后，Session Recording 会关闭该文件并打开一个新文件以继续进行录制。此操作称为滚动。

可以为一个滚动指定两个阈值：

- **文件大小。**文件达到指定的兆字节数之后，Session Recording 会关闭该文件并打开一个新文件。默认情况下，文件在达到 50 MB 之后会发生滚动；但您可以将限制指定为 10 MB 至 1 GB 之间。
- **持续时间。**会话录制达到指定的小时数之后，会关闭该文件并打开一个新文件。默认情况下，录制达到 12 小时之后文件会进行滚动；但您可以将限制指定为 1 至 24 小时之间。

Session Recording 会检查这两个字段，确定先发生哪个事件，从而决定滚动时间。例如，如果指定文件大小为 17 MB 且持续时间为 6 小时，而录制件在 3 小时内达到 17 MB，Session Recording 就会对 17 MB 的文件大小作出反应，关闭该文件并打开一个新文件。

为防止创建许多小文件，Session Recording 会在至少一小时之后才进行滚动，而不考虑文件大小的指定值，这一时间是您可以输入的最小值。文件大小超过 1 GB 属此规则的例外情况。

为文件指定最大限制

1. 登录到托管 Session Recording Server 的计算机。
2. 从开始菜单中，选择 Session Recording Server Properties。
3. 在 Session Recording Server Properties 中，单击滚动选项卡。
4. 输入一个 10 至 1024 之间的整数，以指定文件的最大大小（以 MB 为单位）。
5. 输入一个 1 至 24 之间的整数，以指定录制的最大持续时间（以小时为单位）。

查看录制

May 04, 2016

使用 Session Recording Player 查看和搜索录制的 XenApp 或 XenDesktop 会话，并为这些会话添加书签。

如果录制会话时启用了实时播放功能，则可以查看正在进行的会话（有数秒延迟），也可查看已完成的会话。

如果会话的持续时间较长或文件大小较大，超出 Session Recording 管理员配置的限制，则会显示在多个会话文件中。

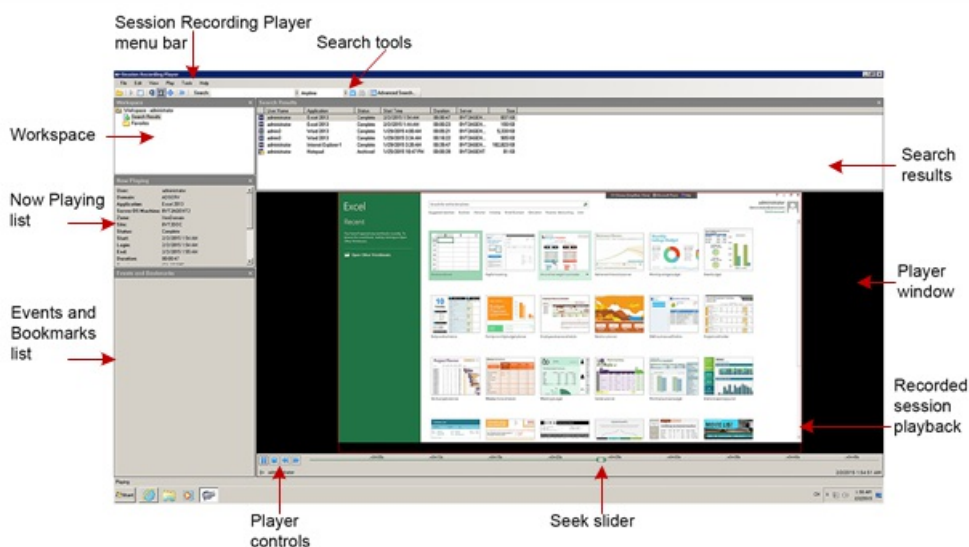
注意：Session Recording 管理员必须授予用户访问录制的服务器操作系统计算机会话的权限。如果您被拒绝访问查看会话，请与 Session Recording 管理员联系。

安装 Session Recording Player 时，Session Recording 管理员通常会设置 Session Recording Player 与 Session Recording Server 之间的连接。如果未设置此连接，首次执行文件搜索时，系统会提示您进行设置。有关设置信息，请与您的 Session Recording 管理员联系。

启动 Session Recording Player

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。
将显示 Session Recording Player。

下图显示的 Session Recording Player 用插图编号指出了其主要元素。以下整篇文章都介绍了这些元素的功能。



显示或隐藏窗口元素

Session Recording Player 包含将打开和关闭的窗口元素。

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。
3. 从 Session Recording Player 菜单栏中，选择查看。
4. 选择要显示的元素。选择某元素之后，该元素会立即显示。复选标记指示该元素处于选中状态。

更改 Session Recording Server

如果 Session Recording 管理员为您的 Session Recording Player 设置了可连接多个 Session Recording Server 的功能，则可以
选择 Session Recording Player 连接到的 Session Recording Server。Session Recording Player 一次只能连接到一个 Session

Recording Server。

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。
3. 从 Session Recording Player 菜单栏中，依次选择工具 > 选项 > 连接。
4. 选择要连接的 Session Recording Server。

打开和播放录制件

May 04, 2016

可以用三种方式打开 Session Recording Player 中的会话录制：

- 使用 Session Recording Player 执行搜索。满足搜索条件的录制的会话将显示在搜索结果区域中。
- 直接从本地磁盘驱动器或共享驱动器访问录制的会话文件。
- 从“收藏夹”文件夹访问录制的会话文件。

打开未使用数字签名录制的文件时，系统会显示一条警告消息，指出未验证该文件的来源和完整性。如果您确信该文件的完整性，请在警告弹出窗口中单击是以打开该文件。

打开和播放搜索结果区域中的记录

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。
3. 执行搜索。
4. 如果未显示搜索结果区域，请在“工作区”窗格中选择搜索结果。
5. 在搜索结果区域中，选择要播放的会话。
6. 执行以下任何操作：
 - 双击会话
 - 单击鼠标右键并选择播放
 - 从 Session Recording Player 菜单栏中，选择播放 > 播放

通过访问文件打开和播放录制件

录制的会话文件名以 i_ 开头，后面依次是唯一的字母数字文件 ID、.icl 和 .icle 文件扩展名：.icl 是未应用播放保护的录制件，而 .icle 是应用了播放保护的录制件。Session Recording 将录制的会话文件保存在包含会话录制日期的目录结构中。例如，在 2014 年 12 月 22 日录制的会话文件，保存在文件夹路径 2014\12\22 下。

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。
3. 执行以下任何操作：
 - 在 Session Recording Player 菜单栏中，依次选择文件 > 打开，并浏览文件
 - 使用 Windows 资源管理器，导航至文件，并将文件拖动到播放器窗口中
 - 使用 Windows 资源管理器，导航至文件，并双击该文件
 - 如果在“工作区”窗格中创建了“收藏夹”，请选择收藏夹，并通过与在搜索结果区域中打开文件相同的方法从“收藏夹”区域打开该文件

使用收藏夹

通过创建“收藏夹”文件夹，您可以快速访问经常查看的录制件。这些“收藏夹”文件夹引用存储在 workstation 或网络驱动器上的录制的会话文件。可以将这些文件导入及导出到其他 workstation，并且可以与其他 Session Recording Player 用户共享这些文件夹。

注意：只有对 Session Recording Player 有访问权限的用户才能下载与“收藏夹”文件夹相关联的录制的会话文件。请与您的 Session Recording 管理员联系获取访问权限。

创建“收藏夹”子文件夹：

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。

3. 在 Session Recording Player 中，选择“工作区”窗格中的收藏夹文件夹。
4. 从菜单栏中，依次选择文件 > 文件夹 > 新建文件夹。收藏夹文件夹下将显示一个新文件夹。
5. 键入文件夹名称，然后按 Enter 或单击任意位置以接受新名称。

使用文件 > 文件夹菜单中显示的其他选项可删除、重命名、移动、复制、导入和导出文件夹。

搜索录制的会话

May 04, 2016

Session Recording Player 允许您执行快速搜索、执行高级搜索，以及指定应用到所有搜索的选项。搜索结果显示在 Session Recording Player 的搜索结果区域。

注意：要显示所有可用的录制会话（搜索中可以显示的会话的最大数量），在执行搜索时请不要指定任何搜索参数。

执行快速搜索

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。
3. 定义搜索条件：
 - 在搜索字段中输入搜索条件。有助于您实现以下操作：
 - 将鼠标指针移动到搜索标签上以显示要用作指导的参数列表
 - 单击搜索字段右侧的箭头以显示最近执行的 64 次搜索的文本
 - 使用搜索字段右侧的下拉列表以选择用于指定会话录制时间的时段或持续时间
4. 单击下拉列表右侧的望远镜图标以启动搜索。

执行高级搜索

提示：高级搜索可能最多需要 20 秒即可返回包含 15 万个以上条目的结果。Citrix 建议使用更精确的搜索条件（例如日期范围或用户）以减少结果数量。

1. 登录到安装了 Session Recording Player 的工作站。
 2. 从开始菜单上选择 Session Recording Player。
 3. 在 Session Recording Player 窗口中，单击工具栏上的高级搜索，或者依次选择工具 > 高级搜索。
 4. 在高级搜索对话框的选项卡中定义搜索条件。
 - 通用允许您按域或帐户授权、站点、组、服务器操作系统计算机、应用程序或文件 ID 进行搜索。
 - 日期/时间允许您搜索日期、星期和一天中的时间。
 - 事件允许您搜索 Session Recording 管理员插入到会话中的自定义事件。
 - 其他允许您按会话名称、客户端名称、客户端地址以及录制持续时间进行搜索。它还允许您为此搜索指定搜索结果的最大显示数目以及搜索中是否包含存档文件。
- 指定搜索条件时，您正在创建的查询将显示在对话框底部的窗格中。
5. 单击搜索以启动搜索。

提示：可以保存和检索高级搜索查询。在高级搜索对话框内单击保存可保存当前的查询。在高级搜索对话框内单击打开可检索已保存的查询。查询另存为扩展名为 .isq 的文件。

设置搜索选项

Session Recording Player 搜索选项允许您限制搜索结果中显示的会话录制件的最大数量，以及指定搜索结果中是否包含存档会话文件。

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。
3. 在 Session Recording Player 菜单栏中，依次选择工具 > 选项 > 搜索。
4. 在要显示的最大结果数量字段中，键入要显示的搜索结果数量。最多可显示 500 条结果。
5. 要设置搜索中是否包含存档文件，请选择或清除包括存档文件。

播放录制的会话

May 04, 2016

在 Session Recording Player 中打开录制的会话之后，可以使用以下方法浏览录制的会话：

- 使用播放器控件播放、停止、暂停以及提高或降低播放速度
- 使用搜寻滑块快进或快退

如果您在录制件中插入了标记，或者录制的会话中包含自定义事件，也可以通过转到这些标记和事件来浏览录制的会话。

注意：

- 在播放录制的会话期间，可能会显示第二个鼠标指针。如果用户在 Internet Explorer 中浏览时单击了一个图像，而该图像的原始大小比屏幕大，但自动被 Internet Explorer 缩小，则会在录制件中显示第二个指针。虽然会话期间仅显示一个指针，但播放过程中可能会显示两个。
- 此 Session Recording 版本不支持适用于 XenApp 的 SpeedScreen Multimedia Acceleration，或适用于 XenApp 的 Flash 品质调节策略设置。如果启用了此选项，播放将显示一个黑色方块。
- 使用 HDX RealTime Optimization Pack for Microsoft Lync 时，Session Recording 无法录制 Lync 网络摄像机视频。

使用播放器控件

可以单击播放器窗口下方的播放器控件，也可以从 Session Recording Player 菜单栏中选择播放来访问这些控件。使用播放器控件可以实现以下操作：

	播放所选会话文件。
	暂停播放。
	停止播放。如果单击停止，然后单击播放，记录将从文件开头重新开始播放。
	将当前播放速度减半，最多可减至正常速度的四分之一。
	将当前播放速度加倍，最多可增加到正常播放速度的 32 倍。

使用搜寻滑块

使用播放器窗口下方的搜寻滑块可跳至录制的会话中的不同位置。可以将搜寻滑块拖动到录制件中要查看的点，也可以单击滚动条的任意位置以移动到该位置。

还可以使用以下键盘快捷键来控制搜寻滑块：

键：	搜寻操作：
主页	搜寻到开头。
End	搜寻到结尾。

右键	向前搜寻 5 秒。
向左键	向后搜寻 5 秒。
将鼠标滚轮向下滚动一格	向前搜寻 15 秒。
将鼠标滚轮向上滚动一格	向后搜寻 15 秒。
Ctrl + 向右键	向前搜寻 30 秒。
Ctrl + 向左键	向后搜寻 30 秒。
Page Down	向前搜寻 1 分钟。
Page Up	向后搜寻 1 分钟。
Ctrl + 将鼠标滚轮向下滚动一格	向前搜寻 90 秒。
Ctrl + 将鼠标滚轮向上滚动一格	向后搜寻 90 秒。
Ctrl + Page Down	向前搜寻 6 分钟。
Ctrl + Page Up	向后搜寻 6 分钟。

注意：调整搜寻滑块的速度：从 Session Recording Player 菜单栏中，依次选择工具 > 选项 > 播放器，并拖动滑块以增加或减少搜寻响应时间。响应时间越快，所需内存越多。响应速度可能会很慢，具体取决于录制内容的大小和计算机的硬件。

更改播放速度

您可以对 Session Recording Player 进行设置，使其能够以正常播放速度的 1/4 到 32 倍逐渐增加的顺序来播放录制的会话。

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。
3. 从 Session Recording Player 菜单栏中，选择播放 > 播放速度。
4. 选择一个速度选项。

速度会立即调整。在播放器窗口控件的下方将显示一个数字，指示增加或减少的速度。指示指数率的文本在播放器窗口中会暂时显示为绿色。

跳过没有操作的空白部分

在快速查看模式下，可以将 Session Recording Player 设置为跳过录制的会话中没有发生操作的部分。此设置可节省观看播放的时间；但是，它不会跳过连续动画，例如连续移动的鼠标指针、闪烁的光标或显示的包含秒针移动的时钟。

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。

3. 从 Session Recording Player 菜单栏中，依次选择播放 > 快速查看模式。

该选项将打开或关闭。每次选择该选项时，其状态在播放器窗口中会暂时显示为绿色。

使用事件和书签

May 04, 2016

可以使用事件和书签帮助浏览录制的会话。

可以使用事件 API 和第三方应用程序在会话录制过程中将事件插入到会话中。事件作为会话文件的一部分进行保存。无法使用 Session Recording Player 删除或修改事件。

书签是使用 Session Recording Player 插入到录制的会话中的标记。删除之前，书签会一直与该录制的会话相关联，但它们不作为会话文件的一部分进行保存。默认情况下，每个书签都标有文本书签，但您可以将其更改为最多包含 128 个字符的任何文本附注。

事件和书签在播放器窗口下方显示为圆点。事件显示为黄色圆点，书签显示为蓝色圆点。将鼠标移动到这些圆点上可显示与其相关联的文本标签。还可以在 Session Recording Player 的事件和书签列表中显示事件和书签。在此列表中，事件和书签与其文本标签及其在录制的会话中显示的次数一起显示（按时间顺序显示）。

可以使用事件和书签帮助浏览录制的会话。通过转到事件或书签，可以跳至录制的会话中事件或书签的插入点。

显示列表中的事件和书签

事件和书签列表显示当前正在播放的录制会话中插入的事件和书签。可以只显示事件、只显示书签，也可以同时显示二者。

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。
3. 将鼠标指针移动到事件和书签列表区域，然后单击鼠标右键以显示菜单。
4. 选择只显示事件、只显示书签或显示全部。

插入书签

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。
3. 开始播放要添加书签的录制会话。
4. 将搜寻滑块移动到要插入书签的位置。
5. 将鼠标指针移动到播放器窗口区域，然后单击鼠标右键以显示菜单。
6. 使用默认标签书签添加书签，或创建附注：
 - 要使用默认标签书签添加书签，请选择添加书签。
 - 要使用您创建的描述性文本标签添加书签，请选择添加附注。键入要指定给该书签的文本标签（最多包含 128 个字符）。单击确定。

添加或更改附注

创建书签后，可以向其添加附注或更改其附注。

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。
3. 开始播放包含书签的录制会话。
4. 确保事件和书签列表可显示书签。
5. 选择事件和书签列表中的书签，然后单击鼠标右键以显示菜单。
6. 选择编辑附注。
7. 在显示的窗口中，键入新附注，然后单击确定。

删除书签

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。
3. 开始播放包含书签的录制会话。
4. 确保事件和书签列表可显示书签。
5. 选择事件和书签列表中的书签，然后单击鼠标右键以显示菜单。
6. 选择删除。

转到事件或书签

转到事件或书签会导致 Session Recording Player 转到录制的会话中事件或书签的插入点。

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。
3. 开始播放包含事件或书签的会话录制件。
4. 转到事件或书签：
 - 在播放器窗口下方的区域中，单击表示事件或书签的圆点以转到事件或书签。
 - 在事件和书签列表中，双击要转到的事件或书签。要转到下一个事件或书签，请选择列表中的任何事件或书签，单击鼠标右键以显示菜单，然后选择搜寻至书签。

更改播放显示

May 04, 2016

通过“选项”，您可以更改录制的会话在播放器窗口中的显示方式。可以平移和缩放图像、以全屏模式显示播放、在单独的窗口中显示播放器窗口，以及在录制的会话周围显示红色边框以与播放器窗口背景进行区分。

以全屏模式显示播放器窗口

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。
3. 从 Session Recording Player 菜单栏中，依次选择查看 > 全屏播放。
4. 要返回原始大小，请按 Esc 或 F11。

在单独的窗口中显示播放器窗口

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。
3. 从 Session Recording Player 菜单栏中，依次选择查看 > 播放器在单独窗口中。将显示一个包含播放器窗口的新窗口。您可以拖动该窗口以及调整其大小。
4. 要将播放器窗口嵌入到主窗口中，请依次选择查看 > 播放器在单独窗口中，或按 F10。

缩放会话播放使其适应播放器窗口

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。
3. 从 Session Recording Player 菜单栏中，依次选择播放 > 平移和缩放 > 缩放以适应窗口。
 - 缩放以适应窗口(快速呈现) 在提供高质量图像的同时会缩小图像。与使用“高质量”选项时相比，图像的绘制速度更快，但图像和文本的锐度要差一些。如果在使用“高质量”模式时遇到性能问题，请使用此选项。
 - 缩放以适应窗口(高质量) 在提供高质量图像和文本的同时会缩小图像。使用此选项可能会导致图像的绘制速度比使用“快速呈现”选项慢得多。

平移图像

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。
3. 从 Session Recording Player 菜单栏中，依次选择播放 > 平移和缩放 > 平移。指针变为手形，并且播放器窗口的右上角会显示屏幕的小图示。
4. 拖动图像。小图示指示您在图像中的位置。
5. 要停止平移，请选择其中一个缩放选项。

在会话录制件周围显示红色边框

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。
3. 在 Session Recording Player 菜单栏中，依次选择工具 > 选项 > 播放器。
4. 选中显示会话录制件周围的边框复选框。

提示：如果未选中显示会话录制件周围的边框复选框，可以在指针位于播放器窗口中时，通过单击并按住鼠标左键暂时显示红色边框。

缓存录制的会话文件

May 04, 2016

每次打开录制的会话文件时，Session Recording Player 都会从录制的存储位置下载该文件。如果您经常下载相同的文件，您可以通过在工作站上缓存文件来节省下载所需的时间。缓存的文件存储在工作站上的以下文件夹中：

userprofile\AppData\Local\Citrix\SessionRecording\Player\Cache

您可以指定用于缓存的磁盘空间量。录制内容填满指定的磁盘空间之后，Session Recording 会删除最旧、使用次数最少的录制文件，从而为新录制文件释放空间。可以随时清空缓存来释放磁盘空间。

启用缓存

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。
3. 从 Session Recording Player 菜单栏中，依次选择工具 > 选项 > 缓存。
4. 选中将下载的文件缓存在本地计算机复选框。
5. 如果要限制用于缓存的磁盘空间量，请选择限制要使用的磁盘空间量复选框，并指定要用于缓存的兆字节数。
6. 单击确定。

清空缓存

1. 登录到安装了 Session Recording Player 的工作站。
2. 从开始菜单上选择 Session Recording Player。
3. 从 Session Recording Player 菜单栏中，依次选择工具 > 选项 > 缓存。
4. 选中将下载的文件缓存在本地计算机复选框。
5. 在 Session Recording Player 中，依次选择工具 > 选项 > 缓存。
6. 单击清除缓存，然后单击确定以确认操作。

Session Recording 故障排除

May 04, 2016

此故障排除信息包含一些解决方案，用于解决安装 Session Recording 组件的过程中以及安装之后可能会遇到的问题：

- 组件无法相互连接
- 无法录制会话
- Session Recording Player 或 Session Recording 策略控制台的问题
- 与通信协议有关的问题

Session Recording Agent 无法连接

Session Recording Agent 无法连接时，将记录 Exception caught while sending poll messages to Session Recording Broker（向 Session Recording Broker 发送轮询消息时捕获到异常）事件消息，后跟异常文本。此异常文本会提供连接失败的原因。这些原因包括：

- 基础连接已关闭。无法为 SSL/TLS 安全通道建立信任关系。此异常表示 Session Recording Server 使用的是 Session Recording Agent 所在的服务器不信任或没有 CA 证书的 CA 签发的证书。或者，证书可能已过期或已被吊销。

解决方案：验证托管 Session Recording Agent 的服务器上是否安装了正确的 CA 证书，或者使用可信 CA。

- 远程服务器返回错误: (403) 已禁止。此为尝试使用 HTTP（非安全协议）进行连接时出现的标准 HTTPS 错误。托管 Session Recording Server 的计算机拒绝连接，因为该计算机仅接受安全连接。

解决方案：使用 Session Recording Agent Properties 将 Session Recording Broker 协议更改为 HTTPS。

Session Recording Broker 在评估录制策略查询时返回未知错误。错误代码 5（访问被拒绝）。有关详细信息，请参阅 Session Recording Server 上的事件日志。启动会话并申请进行录制策略评估时会发生此错误。导致此错误的原因有：从 Session Recording Authorization 控制台的策略查询角色中删除了已通过身份验证的用户组（此为默认成员）。

解决方案：将已通过身份验证的用户组添加回此角色，或将托管每个 Session Recording Agent 的每个服务器添加到策略查询角色中。

基础连接已关闭。服务器关闭了本应保持活动状态的连接。此错误表示 Session Recording Server 已关闭或无法接受请求。这可能是因为 IIS 处于脱机状态或重新启动，或者整个服务器可能处于脱机状态。

解决方案：验证 Session Recording Server 是否已启动、IIS 是否正在该服务器上运行，以及该服务器是否已连接到网络。

Session Recording Server 无法连接到 Session Recording 数据库

当 Session Recording Server 无法连接到 Session Recording 数据库时，可能会显示类似以下内容的错误消息之一：

Event Source:

Citrix Session Recording Storage Manager Description: Exception caught while establishing database connection.（事件源: Citrix Session Recording Storage Manager 描述: 建立数据库连接时捕获到异常。）此错误在托管 Session Recording Server 的计算机的事件查看器中的应用程序事件日志中显示。

Unable to connect to the Session Recording Server. Ensure that the Session Recording Server is running.（无法连接到 Session Recording Server。请确保 Session Recording Server 正在运行。）启动 Session Recording 策略控制台时会显示此错误消息。

解决方案：

- Microsoft SQL Server 2008 R2 Express Edition、Microsoft SQL Server 2012 Express Edition 或 Microsoft SQL Server 2014 Express Edition 安装在独立服务器上，并且为 Session Recording 配置的服务或设置不正确。该服务器必须启用 TCP/IP 协议，并运行 SQL Server Browser 服务。有关启用这些设置的信息，请参阅 Microsoft 文档。
- Session Recording 安装期间（“管理”部分）提供了错误的服务器和数据库信息。卸载 Session Recording 数据库并重新安装，以提供正确的信息。
- Session Recording 数据库服务器处于关闭状态。请验证该服务器是否处于连接状态。
- 托管 Session Recording Server 的计算机或托管 Session Recording 数据库服务器的计算机无法解析对方的 FQDN 或 NetBIOS 名称。请使用 ping 命令验证这些名称是否可解析。

用户“NT_AUTHORITY\ANONYMOUS LOGON”登录失败。此错误消息表示用户以 .\administrator 身份错误地登录了服务。

解决方案：请以本地系统用户身份重新启动服务，并重新启动 SQL 服务。

未录制会话

如果未成功录制应用程序会话，请先检查运行 Session Recording Agent 和 Session Recording Server 的服务器操作系统计算机上事件查看器中的应用程序事件日志。此日志可能会提供有价值的诊断信息。

如果未录制会话，以下问题可能是导致该问题的原因：

- **组件连接和证书。**如果 Session Recording 组件相互之间无法进行通信，可能会导致会话录制失败。要解决录制问题，请验证是否已将所有组件正确配置为指向正确的计算机，以及所有证书是否均有效且已正确安装。
- **非 Active Directory 域环境。**Session Recording 设计为在 Microsoft Active Directory 域环境中运行。如果您不在 Active Directory 环境中运行，可能会遇到录制问题。确保所有 Session Recording 组件都在属于 Active Directory 域成员的计算机上运行。
- **会话共享与活动策略冲突。**Session Recording 将活动策略与用户打开的第一个已发布应用程序相匹配。随后在同一会话中打开的应用程序都将继续遵循对第一个应用程序有效的策略。要防止会话共享与活动策略冲突，请在单独的服务器操作系统计算机上发布冲突的应用程序。
- **未启用录制。**默认情况下，在服务器操作系统计算机上安装 Session Recording Agent 会启用服务器的录制功能。将活动录制策略配置为允许启用此功能之后，才开始录制。
- **活动录制策略不允许录制。**对于要录制的会话，活动录制策略必须允许录制用户、服务器或已发布应用程序的会话。
- **Session Recording 服务未运行。**对于要录制的会话，Session Recording Agent 服务必须在服务器操作系统计算机上运行，并且 Session Recording Storage Manager 服务必须在托管 Session Recording Server 的计算机上运行。
- **未配置 MSMQ。**如果未在运行 Session Recording Agent 的服务器上以及托管 Session Recording Server 的计算机上正确配置 MSMQ，可能会出现录制问题。

无法查看实时会话播放

如果在使用 Session Recording Player 查看录制件时遇到困难，屏幕上可能会显示以下错误消息：

下载录制的会话文件失败。不允许实时会话播放。服务器配置为不允许此功能。此错误表明服务器配置为不允许此操作。

解决方案：在 Session Recording Server Properties 对话框中，选择播放选项卡，然后选择允许实时会话播放复选框。

录制损坏或不完整

使用 Session Recording Player 查看录制内容时，如果录制内容损坏或不完整，您可能还会在 Session Recording Agent 的事件日志中看到警告。

Event Source: Citrix Session Recording Storage Manager

Description: Data lost while recording file （事件源：Citrix Session Recording Storage Manager 描述：录制文件 期间数据丢失）

使用 Machine Creation Services (MCS) 或 Provisioning Services 创建安装了已配置的主映像和 Microsoft Message Queuing (MSMQ) 的 VDA 时通常会出现此问题。在这种情况下，VDA 的 QMId 与 MSMQ 相同。

解决方案：为每个 VDA 创建唯一的 QMId。 [已知问题](#) 中介绍了解决方案。

安装 Session Recording 数据库或 Session Recording Server 时，测试数据库实例的连接性失败。

安装 Session Recording 数据库或 Session Recording Server 时，测试连接失败并显示错误消息 **Database connection test failed. Please correct Database instance name**（数据库连接测试失败。请更正数据库实例名称），即使数据库实例名称正确也是如此。

解决方案：确保当前用户具有公共 SQL Server 角色权限，可以更正权限限制失败。

验证组件连接

May 04, 2016

安装 Session Recording 的过程中，组件可能无法连接至其他组件。所有组件都能与 Session Recording Server (Broker) 通信。默认情况下，Broker（一种 IIS 组件）使用 IIS 默认的 Web 站点证书来确保安全。如果某个组件无法连接至 Session Recording Server，则其他组件在尝试连接时可能也会失败。

Session Recording Agent 和 Session Recording Server（存储管理器和 Broker）会将连接错误记入应用程序事件日志，该日志位于托管 Session Recording Server 的计算机的事件查看器中，而 Session Recording 策略控制台和 Session Recording Player 会在无法连接时在屏幕上显示连接错误消息。

验证 Session Recording Agent 是否已连接

1. 登录到安装了 Session Recording Agent 的服务器。
2. 从开始菜单中，选择 Session Recording Agent Properties。
3. 在 Session Recording Server Properties 中，单击连接。
4. 验证 Session Recording Server 的值是否为托管 Session Recording Server 的计算机的正确服务器名称。
5. 验证服务器操作系统计算机是否可以访问作为 Session Recording Server 的值给定的服务器。

注意：检查应用程序事件日志中是否记录了错误和警告。

验证 Session Recording Server 是否已连接

警告：“注册表编辑器”使用不当会导致严重问题，可能需要重新安装操作系统。Citrix 无法保证因“注册表编辑器”使用不当导致出现的问题能够得以解决。使用“注册表编辑器”需自担风险。

1. 登录到托管 Session Recording Server 的计算机。
2. 打开“注册表编辑器”。
3. 浏览至 HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\SmartAuditor\Server。
4. 验证 SmAudDatabaseInstance 的值是否正确引用了 SQL Server 实例中安装的 Session Recording 数据库。

验证 Session Recording 数据库是否已连接

1. 使用 SQL 管理工具打开包含安装的 Session Recording 数据库的 SQL 实例。
2. 打开 Session Recording 数据库的安全性权限。
3. 验证 Session Recording 计算机帐户是否可以访问数据库。例如，如果托管 Session Recording Server 的计算机在 MIS 域中命名为 SsRecSrv，则应将数据库中的计算机帐户配置为 MIS\SsRecSrv\$。该值是在安装 Session Recording 数据库期间配置的。

测试 IIS 连接

通过使用 Web 浏览器访问 Session Recording Broker Web 页面来测试与 Session Recording Server IIS 站点的连接有助于确定 Session Recording 组件间的通信问题是否来自错误配置的协议配置、证书问题或启动 Session Recording Broker 时出现的问题。

验证 Session Recording Agent 的 IIS 连接

1. 登录到安装了 Session Recording Agent 的服务器。
2. 启动 Web 浏览器，并输入以下地址：
 - 对于 HTTPS：https://servername/SessionRecordingBroker/RecordPolicy.rem?wsdl 其中 servername 是指托管 Session Recording Server 的计算机的名称

- 对于 HTTP : <http://servername/SessionRecordingBroker/RecordPolicy.rem?wsdl>其中 servername 是指托管 Session Recording Server 的计算机的名称
3. 如果系统提示您进行 NT LAN Manager (NTLM) 身份验证, 请使用域管理员帐户进行登录。

如果您在浏览器中看到 XML 文档, 则表明运行 Session Recording Agent 的计算机已通过配置协议连接到托管 Session Recording Server 的计算机。

验证 Session Recording Player 的 IIS 连接

1. 登录到安装了 Session Recording Player 的工作站。
2. 启动 Web 浏览器, 并输入以下地址:
 - 对于 HTTPS : <https://servername/SessionRecordingBroker/Player.rem?wsdl>其中 servername 是指托管 Session Recording Server 的计算机的名称
 - 对于 HTTP : <http://servername/SessionRecordingBroker/Player.rem?wsdl>其中 servername 是指托管 Session Recording Server 的计算机的名称
3. 如果系统提示您进行 NT LAN Manager (NTLM) 身份验证, 请使用域管理员帐户进行登录。

如果您在浏览器中看到 XML 文档, 则表明运行 Session Recording Player 的计算机已通过配置协议连接到托管 Session Recording Server 的计算机。

验证 Session Recording 策略控制台的 IIS 连接

1. 登录到安装了 Session Recording 策略控制台的服务器。
2. 启动 Web 浏览器, 并输入以下地址:
 - 对于 HTTPS : <https://servername/SessionRecordingBroker/PolicyAdministration.rem?wsdl>其中 servername 是指托管 Session Recording Server 的计算机的名称
 - 对于 HTTP : <http://servername/SessionRecordingBroker/PolicyAdministration.rem?wsdl>其中 servername 是指托管 Session Recording Server 的计算机的名称
3. 如果系统提示您进行 NT LAN Manager (NTLM) 身份验证, 请使用域管理员帐户进行登录。

如果您在浏览器中看到 XML 文档, 则表明运行 Session Recording 策略控制台的计算机已通过配置协议连接到托管 Session Recording Server 的计算机。

对证书问题进行故障排除

如果您将 HTTPS 作为通信协议, 则必须使用服务器证书对托管 Session Recording Server 的计算机进行配置。所有组件与 Session Recording Server 的连接均必须具有根证书颁发机构 (CA)。否则, 组件之间的连接尝试将失败。

可以在测试 IIS 连接时根据需要通过访问 Session Recording Broker Web 页面来测试证书。如果能够访问各个组件的 XML 页面, 则证书配置正确。

下面是证书问题导致连接失败的一些常见方式:

- **证书无效或丢失。**如果运行 Session Recording Agent 的服务器没有用于信任服务器证书的根证书, 则无法建立信任, 并通过 HTTPS 连接至 Session Recording Server, 最终导致连接失败。请验证所有组件是否均信任 Session Recording Server 上的服务器证书。
- **命名不一致。**如果分配给托管 Session Recording Server 的计算机的服务器证书是使用完全限定域名 (FQDN) 创建的, 则连接至 Session Recording Server 时, 所有连接的组件均必须使用 FQDN。如果使用的是 NetBIOS 名称, 请通过 Session Recording Server 的 NetBIOS 名称对组件进行配置。
- **证书已过期。**如果服务器的证书已过期, 则通过 HTTPS 与 Session Recording Server 的连接会失败。请验证分配给托管 Session Recording Server 的计算机的服务器证书是否有效且未过期。如果使用同一证书对会话录制进行数字签名, 则托管 Session Recording Server 的计算机的事件日志会提供错误消息, 表明证书已过期, 或在证书即将过期时提供警告信息。

如果 Session Recording Player 失败，搜索录制件

May 04, 2016

如果使用 Session Recording Player 搜索录制件时遇到困难，屏幕上可能显示以下错误消息：

- 搜索录制的会话文件失败。无法解析远程服务器名称: servername。其中 servername 是 Session Recording Player 尝试连接到的服务器名称。Session Recording Player 无法访问 Session Recording Server。出现此种情况有两种可能原因：键入了错误的服务器名称，或者 DNS 无法解析服务器名称。
解决方案：从播放器菜单栏中，依次选择工具 > 选项 > 连接，并验证 Session Recording Servers 列表中列出的服务器名称是否正确。如果正确，请在命令提示窗口下运行 ping 命令，以查看是否可解析该名称。Session Recording Server 处于关闭或脱机状态时，搜索录制的会话文件失败，错误消息为无法访问远程服务器。
- 无法访问远程服务器。Session Recording Server 处于关闭或脱机状态时会发生此错误。
解决方案：验证 Session Recording Server 是否已连接。
- 访问被拒绝错误。如果用户未被授予搜索和下载录制的会话文件的权限，可能会发生访问被拒绝错误。
解决方案：使用 Session Recording Authorization 控制台向用户分配播放器角色。
- 搜索录制的会话文件失败。基础连接已关闭。无法为 SSL/TLS 安全通道建立信任关系。导致此异常的原因为：Session Recording Server 使用的是客户端设备不信任或没有 CA 证书的 CA 签发的证书。
解决方案：在安装了 Session Recording Player 的工作站上安装正确或可信的 CA 证书。
- 远程服务器返回错误: (403) 已禁止。此为尝试使用 HTTP（非安全协议）进行连接时出现的标准 HTTPS 错误。服务器拒绝连接，因为默认情况下，将其配置为仅接受安全连接。
解决方案：从 Session Recording Player 菜单栏中，依次选择工具 > 选项 > 连接。从 Session Recordings Servers 列表选择服务器，然后单击修改。将协议从 HTTP 更改为 HTTPS。

MSMQ 故障排除

如果用户看到通知消息，但在 Session Recording Player 中执行搜索后无法找到相关录制件，MSMQ 可能存在问题。请验证队列是否已连接到 Session Recording Server（存储管理器），并使用 Web 浏览器测试是否存在连接错误（如果您将 HTTP 或 HTTPS 用作 MSMQ 通信协议）。

验证队列是否已连接：

1. 登录到托管 Session Recording Agent 的服务器，然后查看传出队列。
2. 验证到托管 Session Recording Server 的计算机的队列是否具有连接状态。
 - 如果状态为“正在等待连接”，队列中有多条消息，且协议为 HTTP 或 HTTPS（与在 Session Recording Agent Properties 对话框的连接选项卡中所选的协议相对应），请执行步骤 3。
 - 如果状态为“已连接”且队列中没有消息，则托管 Session Recording Server 的服务器可能存在问题。请跳过步骤 3 并执行步骤 4。
3. 如果队列中有多条消息，请启动 Web 浏览器并键入以下地址：
 - 对于 HTTPS：https://servername/msmq/private\$/CitrixSmAudData 其中 servername 是指托管 Session Recording Server 的计算机的名称。
 - 对于 HTTP：http://servername/msmq/private\$/CitrixSmAudData 其中 servername 是指托管 Session Recording Server 的计算机的名称。

如果此页面返回一个错误（例如该服务器仅接受安全连接），请将 Session Recording Agent Properties 对话框中列出的 MSMQ 协议更改为 HTTPS。反之，如果此页面报告 Web 站点的安全证书存在问题，SSL/TLS 安全通道的信任关系可能存在问题。在这种情况下，请安装正确的 CA 证书，或使用可信 CA。

4. 如果队列中没有消息，请登录到托管 Session Recording Server 的计算机，并查看专用队列。选择 citrixmaudata。如果队列中有多条消息（消息列的数目），请验证 Session Recording StorageManager 服务是否已启动。如果未启动，请重新启动服务。

更改通信协议

May 04, 2016

出于安全原因，Citrix 建议不要将 HTTP 用作通信协议。Session Recording 安装配置为使用 HTTPS。如果希望使用 HTTP 来代替 HTTPS，必须更改多项设置。

将 HTTP 用作通信协议

1. 登录到托管 Session Recording Server 的计算机，然后禁用 IIS 中的 Session Recording Broker 安全连接。
2. 在每个 Session Recording Agent Properties 对话框中将协议设置从“HTTPS”更改为“HTTP”：
 1. 登录到安装了 Session Recording Agent 的每台服务器。
 2. 从开始菜单中，选择 Session Recording Agent Properties。
 3. 在 Session Recording Agent Properties 中选择连接选项卡。
 4. 在 Session Recording Broker 区域中，从协议下拉列表中选择 HTTP，然后选择确定以接受所做的更改。如果系统提示重新启动服务，请选择是。
3. 在 Session Recording Player 设置中将协议设置从“HTTPS”更改为“HTTP”：
 1. 登录到安装了 Session Recording Player 的每个工作站。
 2. 从开始菜单上选择 Session Recording Player。
 3. 从 Session Recording Player 菜单栏中，依次选择工具 > 选项 > 连接，然后选择该服务器并选择修改。
 4. 从协议下拉列表中选择 HTTP，然后单击确定两次以接受所做的更改，并退出该对话框。
4. 在 Session Recording 策略控制台中将协议设置从“HTTPS”更改为“HTTP”：
 1. 登录到安装了 Session Recording 策略控制台的服务器。
 2. 从开始菜单中，选择 Session Recording 策略控制台。
 3. 从协议下拉列表中选择 HTTP，然后选择确定以进行连接。如果连接成功，下次启动 Session Recording 策略控制台时系统会记住此设置。

恢复为将 HTTPS 用作通信协议

1. 登录到托管 Session Recording Server 的计算机，然后启用 IIS 中的 Session Recording Broker 安全连接。
2. 在每个 Session Recording Agent Properties 对话框中将协议设置从“HTTP”更改为“HTTPS”：
 1. 登录到安装了 Session Recording Agent 的每台服务器。
 2. 从开始菜单中，选择 Session Recording Agent Properties。
 3. 在 Session Recording Agent Properties 中选择连接选项卡。
 4. 在 Session Recording Broker 区域中，从协议下拉列表中选择 HTTPS，然后选择确定以接受所做的更改。如果系统提示重新启动服务，请选择是。
3. 在 Session Recording Player 设置中将协议设置从“HTTP”更改为“HTTPS”：
 1. 登录到安装了 Session Recording Player 的每个工作站。
 2. 从开始菜单上选择 Session Recording Player。
 3. 从 Session Recording Player 菜单栏中，依次选择工具 > 选项 > 连接，然后选择该服务器并选择修改。
 4. 从协议下拉列表中选择 HTTPS，然后单击确定两次以接受所做的更改，并退出该对话框。
4. 在 Session Recording 策略控制台中将协议设置从“HTTP”更改为“HTTPS”：
 1. 登录到安装了 Session Recording 策略控制台的服务器。
 2. 从开始菜单中，选择 Session Recording 策略控制台。
 3. 从协议下拉列表中选择 HTTPS，然后选择确定以进行连接。如果连接成功，下次启动 Session Recording 策略控制台时系统会记住此设置。

参考：管理您的数据库记录

May 04, 2016

ICA 日志数据库 (ICLDB) 实用程序是用于管理会话录制数据库记录的数据库命令行实用程序。此实用程序在 Session Recording 安装期间安装在托管 Session Recording Server 软件的服务器的驱动器:\Program Files\Citrix\SessionRecording\Server\Bin 目录中。

快速参考表

下表列出了 ICLDB 实用程序可用的命令和选项。使用以下格式键入命令：

icldb[version|locate|dormant|import|archive|remove|removeall]命令选项[/l[/f[/s]]/?]

注意：与此实用程序关联的帮助中提供了更多详细说明。要访问帮助，请在命令提示窗口下键入驱动器:\Program Files\Citrix\SessionRecording\Server\Bin 目录，然后键入icldb /?。要访问特定命令的帮助，请键入icldb 命令 /?。

命令	说明
archive	存档超过指定的保留期的会话录制文件。 此命令用于存档文件。
dormant	显示或计数被视为休眠的会话录制文件。休眠文件是指由于数据丢失而未完成的会话录制。 此命令用于在您怀疑有数据丢失时进行验证。可以验证是整个数据库的会话录制文件都进入休眠，还是只有指定天数、小时数或分钟数内录制件进入休眠。
import	将会话录制文件导入 Session Recording 数据库。 此命令用于在丢失数据库录制时重建数据库。 此外，此命令还用于合并数据库（如果您有两个数据库，则可以导入其中一个数据库中的文件）。
locate	将会话录制文件 ID 用作条件查找并显示该文件的完整路径。 查找会话录制文件的存储位置时可使用此命令。 此命令还可用于通过特定文件来验证数据库是否为最新。
remove	从数据库中删除对会话录制文件的引用。 此命令用于清理数据库（使用时需谨慎）。指定要用作条件的保留期。 也可以删除关联的物理文件。
removeall	从 Session Recording 数据库中删除对会话录制文件的所有引用，并将数据库返回原始状态。不会删除实际物理文件；但您无法在 Session Recording Player 中搜索这些文件。 此命令用于清理数据库（使用时需谨慎）。只能通过从备份中恢复来还原删除的引用。
version	显示 Session Recording 数据库架构版本。

命令	说明
	将结果和错误记录到 Windows 事件日志。
/f	强制运行命令而不提示。
/s	不显示版权信息。
/?	显示命令的帮助。

第三方声明

May 04, 2016

Session Recording 可能包含根据以下条款许可的第三方软件组件。此列表是在截止到列出之日使用第三方软件生成的。此列表可能会变更具体的产品版本并且可能不完整；此列表“按原样”提供。在适用法律允许的范围内，CITRIX 或其供应商对于此列表或其准确性或完整性，或使用或分发此列表造成的结果概不负责，也不做任何明示、暗示、法定或其他形式的担保。使用或分发此列表表示您同意，在任何情况下，CITRIX 对于因使用或分发此列表造成的任何特殊的、直接的、间接的或后果性的损害或任何其他损害不承担任何法律责任。

MMC .NET Library

根据 Common Public License 1.0 版许可。

Personal vDisk

May 04, 2016

可以使用诊断工具监视用户对其个人虚拟磁盘的用户数据部分和应用程序部分进行的更改。这些更改包括用户已安装的应用程序和它们修改的文件。更改存储在一组报告中。

1. 在要监视的计算机上，运行 C:\Program Files\Citrix\personal vDisk\bin\CtxPvdDiag.exe。
2. 浏览到要存储报告和日志的位置，选择要生成的报告，然后单击确定。可用报告包括：

报告或日志	生成的文件	监视的更改
软件配置单元报告	Software.Dat.Report.txt、 Software.Dat.delta.txt	Software.Dat.Report.txt 记录用户对 HKEY_LOCAL_MACHINE\Software 配置单元所做的更改。它包括以下部分： • 在基础上安装的应用程序的列表 — 在 0 层上安装的应用程序。 • 用户安装的软件的列表 — 用户在虚拟磁盘的应用程序部分安装的应用程序。 • 用户卸载的软件的列表 — 用户删除的之前存在于 0 层上的应用程序。 有关 Software.Dat.delta.txt 的信息，请参阅配置单元增量报告。
系统配置单元报告	SYSTEM.CurrentControlSet.DAT.Report.txt	此文件记录用户对 HKEY_LOCAL_MACHINE\System 配置单元所作的更改。它包括以下部分： • 用户安装的服务的列表 — 用户安装的服务和驱动程序。 • 更改了以下服务的启动 — 用户更改了启动类型的服务和驱动程序。
安全配置单元报告	SECURITY.DAT.DAT.Report.txt	此文件监视用户在 HKEY_LOCAL_MACHINE\Security 配置单元中所做的更改。
安全帐户管理器 (SAM) 配置单元报告	SAM.DAT.DAT.Report.txt	此文件监视用户在 HKEY_LOCAL_MACHINE\SAM 配置单元中所做的所有更改。
配置单元增量报告	Software.Dat.delta.txt	此文件记录用户在 HKEY_LOCAL_MACHINE\Software 配置单元中添加或删除的所有注册表项和值以及修改的所有值。
Personal vDisk 日志	Pud-IvmSupervisor.log、 PvDActivation.log、PvDSvc.log、 PvDWMI.log、SysVol-IvmSupervisor.log、 vDeskService- <i><#></i> .log	默认情况下，这些文件在 P:\Users\<用户帐户>\AppData\Local\Temp\PVDLOGS 中生成，但被移动到选定的位置。

Windows 操作系统 日志	生成的文件 EvtLog_App.xml、EvtLog_System.xml、 setupapi.app.log、setuperr.log、 setupapi.dev.log、msinfo.txt	监视的更改 EvtLog_App.xml 和 EvtLog_System.xml 是来自个人虚拟磁盘卷的 XML 格式的应用程序和系统事件日志。 Setupapi.app.log 和 setuperr.log 包含 Personal vDisk 安装期间自 msixec.exe 开始运行起生成的日志消息。 Setupapi.dev.log 包含设备安装日志消息。 Msinfo.txt 包含 msinfo32.exe 的输出。有关此输出的信息，请参阅 Microsoft 文档。
文件系统 报告	FileSystemReport.txt	此文件记录用户对文件系统所做的更改。它包括以下部分： ● 重定位的文件 — 用户由 0 层移至虚拟磁盘中的文件。0 层文件是由个人虚拟磁盘所连接的计算机从主映像所继承的文件。 ● 删除的文件 — 通过用户的操作（例如，删除应用程序）隐藏的 0 层文件。 ● 添加的文件（MOF、INF、SYS） — 用户添加到虚拟磁盘的带有 .mof、.inf 或 .sys 扩展名的文件，例如，当用户安装一个通过注册 .mof 文件实现自动恢复功能的应用程序（如 Visual Studio 2010）时。 ● 添加的其他文件 — 用户添加到虚拟磁盘的其他文件，例如，当用户安装应用程序时。 ● 修改但未重定位的基本文件 — 已由用户修改但个人虚拟磁盘内核模式驱动程序未在虚拟磁盘中捕捉到的 0 层文件。

配置日志记录

May 04, 2016

配置日志记录捕获针对数据库的站点配置更改和管理活动。您可以使用记录的内容进行以下操作：

- 在发生配置更改后诊断问题和故障排除；日志提供导航路径记录
- 协助变更管理及跟踪配置
- 报告管理活动

您可以设置配置日志记录首选项，显示配置日志，并从 Citrix Studio 生成 HTML 和 CSV 报告。您可以按日期范围和全文搜索结果过滤显示的配置日志。如果启用强制日志记录，可以阻止进行配置更改，除非这些更改可以记入日志。只要具有适当的权限，即可删除配置日志中的条目。您无法使用配置日志记录功能编辑日志内容。

配置日志记录使用 PowerShell 2.0 SDK 和 Configuration Logging Service。Configuration Logging Service 会在站点中的每个 Controller 上运行；如果某个 Controller 出现故障，另一个 Controller 上的服务将自动处理日志记录请求。

默认情况下，配置日志记录功能处于启用状态，它使用创建站点时所创建的数据库（站点配置数据库）。Citrix 强烈建议您在创建站点后尽快更改配置日志记录数据库的位置。配置日志记录数据库与站点配置数据库支持相同的高可用性功能。

对配置日志记录的访问通过委派管理进行控制，需要具有编辑日志记录首选项和查看配置日志权限。

配置日志会在创建时进行本地化。例如，以英语创建的日志将以英语显示，而无论阅读器的区域设置如何。

通过 Studio、Director 和 PowerShell 脚本启动的配置更改和管理活动都在记录范围之内。记录的配置更改包括对以下项目的处理（创建、编辑、删除和分配）：

- 计算机目录
- 交付组（包括更改电源管理设置）
- 管理员角色和作用域
- 主机资源和连接
- 通过 Studio 管理的 Citrix 策略

记录的管理更改示例包括：

- 虚拟机或用户桌面的电源管理
- Studio 或 Director 向用户发送消息

以下操作不在记录范围之内：

- 自动操作，如虚拟机的池管理启动。
- 通过组策略管理控制台 (GPMC) 实施的策略操作；使用 Microsoft 工具查看这些操作的日志。
- 通过注册表和直接访问数据库进行的更改，或来自 Studio、Director 或 PowerShell 以外的源的更改。
- 初始化部署后，配置日志记录从在 Configuration Service 中注册首个 Configuration Logging Service 实例时开始可用。因此，早期阶段的配置不会记入日志（例如，获取和应用数据库架构以及初始化虚拟机管理程序期间的更改）。

管理配置日志记录

May 04, 2016

默认情况下，配置日志记录使用在您创建站点时所创建的数据库（也称为站点配置数据库）。Citrix 强烈建议您在创建站点后更改用于配置日志记录的数据库（以及用于 Monitoring Service 的数据库，Monitoring Service 默认情况下也会使用站点配置数据库）的位置，原因如下：

- 配置日志记录数据库的备份策略可能与站点配置数据库的备份策略有所不同。
- 通过配置日志记录（以及 Monitoring Service）收集的数据量可能会对站点配置数据库的可用空间造成负面影响。
- 它会针对三个数据库拆分单点故障。

注意：不支持配置日志记录的产品版本在 Studio 中没有日志记录节点。有关详细信息，请参阅 [XenDesktop 7.6](#) 和 [XenApp 7.6 功能及授权](#)。

默认情况下，配置日志记录处于启用状态，强制日志记录处于禁用状态。

1. 在 Studio 导航窗格中，选择日志记录。
2. 在“操作”窗格中选择首选项。“配置日志记录”对话框中包含数据库信息，并指示配置日志记录和强制日志记录处于启用还是禁用状态。
 - 要启用配置日志记录，请选择启用日志记录单选按钮。此为默认设置。如果无法向数据库写入信息，则日志记录信息将被丢弃，但操作仍继续。
 - 要禁用配置日志记录，请选择禁用日志记录单选按钮。如果先前已启用日志记录，现有的日志仍然可通过 PowerShell SDK 进行读取。
 - 要启用强制日志记录，请清除数据库断开连接时允许更改复选框。通常将记入日志的配置更改或管理活动将不被允许，除非可将其写入配置日志记录数据库。
仅当启用配置日志记录，即选择“Enable Configuration Logging”（启用配置日志记录）单选按钮时，才能启用强制日志记录。如果 Configuration Logging Service 出现故障，并且未使用高可用性，则会使用强制日志记录。在这种情况下，将不会执行通常会记入日志的操作。
 - 要禁用强制日志记录，请选中数据库断开连接时允许更改复选框。即使配置日志记录的数据库无法访问，也允许执行配置更改和管理活动。此为默认设置。

注意：启用强制日志记录时无法更改数据库位置，因为更改位置时会断开连接一小段时间，在此期间无法进行日志记录。

1. 使用支持的 SQL Server 版本创建数据库服务器。
2. 在 Studio 导航窗格中，选择日志记录。
3. 在“操作”窗格中选择首选项。
4. 在“日志记录首选项”对话框中，选择更改日志记录数据库。
5. 在“更改日志记录数据库”对话框中，指定包含新数据库服务器的服务器位置（使用下表中的格式之一）以及数据库名称。

数据库类型	输入内容	使用此数据库配置
独立或镜像	服务器名称	使用默认实例，SQL Server 使用默认端口。
	服务器名称\实例名称	使用指定的实例，SQL Server 使用默认端口。

数据库类型	服务器名称,端口号 输入内容	使用默认实例。SQL Server 使用自定义端口。（需要逗号。） 使用此数据库配置
其他	群集名称	群集数据库。
	可用性组侦听器	始终开启的数据库。

- 要允许 Studio 创建数据库，请单击确定。出现提示时，单击确定，系统将自动创建数据库。Studio 会尝试使用当前 Studio 用户的凭据访问数据库；如果此操作失败，系统会提示您输入数据库用户的凭据。然后，Studio 会将数据库架构上载到数据库。（凭据只会在创建数据库期间保留。）
- 要手动创建数据库，请单击生成数据库脚本。生成的脚本包括有关手动创建数据库的说明。在上载架构之前，请确保数据库为空，且至少有一个用户有权访问并更改该数据库。

先前数据库中的配置日志记录数据不会导入新数据库中。检索日志时，不能合并来自两个数据库的日志。新配置日志记录数据库中的第一个日志条目将指明发生了数据库更改，但无法确定先前的数据库。

启动配置更改和管理活动时，Studio 的中上部窗格中将显示 Studio 和 Director 创建的高级别操作。高级别操作会导致一个或多个服务和 SDK 调用，后者为低级别操作。在中上部窗格中选择一项高级别操作时，中下部窗格将显示低级别操作。

如果操作在完成之前失败，可能无法在数据库中完成日志操作；例如，开始记录将没有对应的停止记录。在这种情况下，日志会指出缺少信息。在基于时间范围显示日志时，如果不完整日志中的数据符合条件，则会显示这些不完整的日志。例如，当请求过去五天的所有日志时，如果存在的某个日志的开始时间在过去五天内但没有结束时间，则会包括该日志。

在使用脚本调用 PowerShell cmdlet 时，如果您在创建低级别操作时不指定高级别父操作，则配置日志记录将创建替代的高级别操作。

要显示配置日志内容，请在 Studio 导航窗格中选择日志记录。默认情况下，在中心窗格显示的内容会按时间顺序列出日志内容（最新的条目在最前面），并按日期进行分隔。

显示内容过滤 条件	完成此操作
搜索结果	在中间窗格顶部的搜索框中输入文本。过滤的显示内容包括搜索结果的数量。要返回到标准的日志记录显示，请清除搜索框中的文本。
列标题	单击列标题可以按该字段对显示内容排序。
日期范围	从中间窗格顶部搜索框旁边的下拉列表框中选择一个时间间隔。

您可以生成包含配置日志数据的 CSV 和 HTML 报告。

- CSV 报告包含指定时间间隔内的所有日志记录数据。数据库中的分层数据被简化为单个 CSV 表。所有数据项在此文件中都不具有优先级。不进行任何格式化，也不假定具有可读性。文件（名称为 MyReport）只包含通用格式的数据。CSV 文件通常用于存档数据，或作为报告或数据操作工具（如 Microsoft Excel）的数据源。
- HTML 报告以便于用户理解的格式提供指定时间间隔内的日志记录数据。它提供层次分明的导航视图，便于检查更改。HTML 报告包括两个文件，名称分别为“摘要”和“详细信息”。“摘要”列出了高级别操作：每个操作发生的时间、执行者和结

果。单击每个操作旁边的详细信息链接可转至提供其他信息的“详细信息”文件中的低级别操作。

要生成配置日志报告，请在 Studio 导航窗格中选择日志记录，然后在“操作”窗格中选择 Create custom report（创建自定义报告）。

- 选择报告的日期范围。
- 选择报告格式：CSV、HTML 或二者。
- 浏览到报告的保存位置。

要删除配置日志，必须具有特定的委派管理和 SQL Server 数据库权限。

- **委派管理** — 必须具有允许读取部署配置的委派管理角色。内置的完全权限管理员角色具有此权限。自定义角色必须具有在“其他权限”类别中选择的“只读”或“管理”权限。

要在删除配置日志记录数据之前为其创建备份，自定义角色还必须具有在“日志记录权限”类别中选择的“只读”或“管理”权限。

- **SQL Server 数据库** — 必须具备拥有可从数据库中删除记录权限的 SQL Server 登录。有两种方式实现此要求：
 - 使用具有 sysadmin 服务器角色的 SQL Server 数据库登录名，该角色允许在数据库服务器上执行任何活动。此外，serveradmin 或 setupadmin 服务器角色还允许执行删除操作。
 - 如果部署需要更高的安全性，请使用映射到具有从数据库中删除记录权限的数据库用户的非 sysadmin 数据库登录名。
 1. 在 SQL Server Management Studio 中，以“sysadmin”以外的服务器角色创建 SQL Server 登录名。
 2. 将登录名映射到数据库中的某个用户；SQL Server 将自动在数据库中以登录名创建用户。
 3. 在数据库角色成员身份中，为数据库用户至少指定一种角色成员身份：ConfigurationLoggingSchema_ROLE 或 dbowner。

有关更多信息，请参阅 SQL Server Management Studio 文档。

要删除配置日志，请执行以下操作：

1. 在 Studio 导航窗格中，选择日志记录。
2. 在“操作”窗格中选择删除日志。
3. 在删除日志前，系统会询问是否要创建日志备份。如果选择创建备份，请浏览到应保存备份存档的位置。备份将以 CSV 文件格式创建。

在清除配置日志后，日志删除是发布到空日志的第一项活动。该条目将提供有关删除日志的用户以及时间的详细信息。

Monitor Service OData API

May 04, 2016

除使用 Citrix Director 控制台显示历史数据外，还可以使用 Monitor Service API 查询数据。您可以使用 API 执行以下操作：

- 分析历史趋势以便进行未来规划
- 对连接失败和计算机故障进行细致的故障排除
- 提取信息以用于其他工具和流程；例如，可以使用 Microsoft Excel 的 PowerPivot 表以不同的方式显示数据
- 基于 API 提供的数据构建自定义用户界面

Monitor Service API 使用开放数据协议 (OData)。此协议是一个用于查询和更新数据的 Web 协议，并基于诸如 HTTP 之类的 Web 技术而构建。有关 OData 协议的详细信息，请参阅 <http://www.odata.org>。

Monitor Service API 在使用 Windows Communication Foundation (WCF) Data Services（在处理和合并期间填充）的 SQL Server 数据库的基础上构建。通过 wsHttpBinding 显示了两个使用 WCF 的端点。基址为 `http://{dc-host}/Citrix/Monitor/OData/v2`。还可以使用 SSL 保护端点；有关详细信息，请参阅[使用 SSL 保护端点安全](#)。

1. 数据端点可显示对数据库实体的直接只读访问权限，并且可以使用 OData 查询语言进行访问。根据过滤和列选择，本端点允许高度灵活的访问。数据 API URI 为 `http://{dc-host}/Citrix/Monitor/OData/v2/Data`。有关访问 Monitor Service 数据的详细信息，请参阅[使用 API 访问数据](#)。
2. 方法端点可显示 Citrix Director 用于检索需要复杂分组和高性能标准的数据的服务操作（例如“控制板”和“趋势”页上的查询）。方法 API URI 为 `http://{dc-host}/Citrix/Monitor/OData/v2/Methods`。方法仅在 Director 本身中使用，因而不供大多数 Citrix 客户使用。所以，此处并未介绍这些方法。

XenApp 和 XenDesktop 7.6 附带的 API 版本提供以下新功能：

- **修补程序清单。**在 Director 中使用“用户详细信息”视图或“计算机”视图，您会看到计算机上安装的所有 Citrix 修补程序的列表。您可以使用 API 提取此数据并创建自定义报告（例如，整个站点上安装的修补程序的状态）或将其拉到分析引擎中。引入了新类并且扩展了 Machine 类，可支持跟踪控制器和 VDA 上安装的 Citrix 修补程序。
- **匿名会话故障排除。**会话可作为一组池本地用户帐户运行。API 向 Session 类添加了新属性 IsAnonymous（默认值为 FALSE）。
- **托管应用程序使用率报告。**Director 提供新的容量报告，可显示托管应用程序随时间变化的使用率。API 允许您报告用户会话中运行的每个应用程序实例的详细信息。

以下位置的《API 参考》中完整介绍了所有数据更新：<http://support.citrix.com/help/monitorserviceapi/7.6/>。

本版本已弃用 GetSessionSummary 方法。

使用 API 访问数据

May 04, 2016

以下类型的数据可通过 Monitor Service API 获得：

- 与连接故障相关的数据
- 处于故障状态的计算机
- 会话使用情况
- 登录持续时间
- 负载均衡数据
- 应用于计算机的修补程序
- 托管应用程序的使用情况

有关数据对象的完整说明，请参阅 <http://blogs.citrix.com/2013/08/27/xendesktop-7-monitor-service-what-data-is-available/>。

要使用 Monitor Service OData API，您必须是 XenApp 或 XenDesktop 管理员。要调用 API，您需要具有只读权限；但是，返回的数据由 XenApp 或 XenDesktop 管理员角色和权限决定。例如，交付组管理员可以调用 Monitor Service API，但他们可以获取的数据由使用 Citrix Studio 设置的交付组访问权限控制。有关 XenApp 或 XenDesktop 管理员角色和权限的详细信息，请参阅[委派管理](#)。

查询数据

Monitor Service API 是一款基于 REST 的 API，可以使用 OData 使用者程序进行访问。OData 使用者程序是占用使用 OData 协议显示的数据的应用程序。从简单的 Web 浏览器到可利用 OData 协议的所有功能的自定义应用程序，OData 使用者程序的复杂程度各不相同。有关 OData 使用者程序的详细信息，请参阅：<http://www.odata.org/ecosystem#consumers>。

每一部分 Monitor Service 数据模型都可访问，并且可根据 URL 过滤。OData 以 URL 格式提供查询语言，可用于从服务检索条目。有关详细信息，请参阅：<http://msdn.microsoft.com/en-us/library/ff478141.aspx>

查询在服务器端进行处理，并且可以使用客户端上的 OData 协议进行进一步过滤。

建模数据分为以下三类：聚合数据（汇总表）、对象（计算机、会话等）的当前状态和日志数据，日志数据实际上是指历史事件（如连接）。

注意：OData 协议不支持枚举；在其位置使用整数。要确定 Monitor Service OData API 返回的值，请参阅 <http://support.citrix.com/help/monitorserviceapi/7.6/>。

数据值聚合

Monitor Service 收集多种数据，其中包括用户会话使用情况、用户登录性能详细信息、会话负载均衡详细信息，以及连接和计算机故障信息。根据其类别，数据以不同的方式聚合。了解使用 OData Method API 提供的数据值的聚合是解释数据的关键。例如：

- 一段时间内发生的连接会话故障和计算机故障，因此它们显示为一段时间内的最大值。
- 登录持续时间是时间长度的度量，因此它们显示为一段时间内的平均值。
- 登录计数和连接故障是一段时间内这类事件的计数，因此它们显示为一段时间内的总数。

并发数据评估

会话必须重叠才能视为并发。但是，当时间间隔为 1 分钟时，该时间内的所有会话（无论会话是否重叠）都将被视为并发：由

于时间间隔太小，计算精确度时所涉及的性能开销有些得不偿失。如果会话发生在同一小时，但不同分钟内，则不会被视为重叠。

关联汇总表和行数据

数据模型以两种不同方式表示指标：

- 汇总表以每分钟、每小时和每天的时间粒度表示指标的聚合视图。
- 行数据表示单个事件或在会话、连接、应用程序或其他对象中跟踪到的当前状态。

尝试跨 API 调用或在数据模型自身内部关联数据时，必须理解以下概念和限制：

- **不存在部分时间间隔的汇总数据。** 指标汇总是为了洞察长时间内的历史趋势。这些指标应该聚合到完整时间间隔的汇总表中。不存在仅包含数据收集的开始时间（最早的可用数据）而不包含结束时间的部分时间间隔的汇总数据。这意味着，当查看一天（时间间隔=1440）的聚合时，最早和最近的不完整日期将不包含数据。尽管存在这些部分时间间隔的行数据，但不会汇总这些行数据。对于特定时间粒度，可以通过从特定汇总表提取最小和最大 SummaryDate，确定最早和最近的聚合时间间隔。SummaryDate 列表示时间间隔的开始时间。Granularity 列表示聚合数据的时间间隔长度。
- **按时间关联。** 如上所述，指标聚合到完整时间间隔的汇总表中。它们可以用于了解历史趋势，但是行事件的状态可能更新，不能通过汇总进行趋势分析。基于时间比较汇总数据和行数据时，应注意不存在可能发生的任何部分时间间隔或时间段的开头和结尾部分的汇总数据。
- **缺失的事件和延迟事件。** 如果在聚合时间段有事件缺失或延迟，聚合到汇总表中的指标可能会略有误差。尽管 Monitor Service 尝试维护准确的最新状态，但是它不会针对缺失或延迟的事件后退到过去重新计算汇总表中的聚合值。
- **连接高可用性。** 在连接 HA 期间，当前连接的汇总数据计数会存在误差，但是会话实例仍在行数据中运行。
- **数据保留期限。** 汇总表中的数据保留整理计划不同于行事件数据的计划。由于数据已从汇总表或行表格加以整理，因此可能会有所缺失。不同粒度的汇总数据的保留期限也有所差异。粒度较低的数据（分钟）的整理速度高于粒度较高的数据（天）。如果由于整理导致数据在某个粒度缺失，可以在更高的粒度找到此数据。由于 API 调用仅返回所请求的指定粒度，接收不到某个粒度的数据并不表示同一时间段的数据在更高的粒度上也不存在。
- **时区。** 指标采用 UTC 时间戳存储。汇总表按照小时时区边界聚合。对于没有位于小时边界上的时区，数据聚合的时间可能会有所差异。

数据粒度和保留

Director 检索的聚合数据粒度是所请求的时间 (T) 跨度的函数。规则如下：

- 0 < T <= 1 小时采用每分钟粒度。
- 0 < T <= 30 天采用每小时粒度。
- T > 31 天采用每天粒度。

非来源于聚合数据的请求数据来源于 Session and Connection 行信息。此数据往往增长很快，因此具有自己的整理设置。通过整理可确保仅长期保留相关数据。这样可以确保实现更好的性能，同时维护报告所需的粒度。对于未获得使用 Platinum Edition 许可的客户，不管默认整理保留期限如何，均从第 8 天开始整理。Platinum 客户可以将整理保留期限更改为他们所需的保留天数，如未修改，将使用默认值。

采用以下设置控制整理：

设置名称	受影响的整理	默认值 (天)	访问方法
GroomSessionsRetentionDays	Session 和 SessionDetail 记录	对于非 Platinum 用户为 7；对于	Cmdlet (set/get-monitorconfiguration)

设置名称	受影响的整理	Platinum 默认值 为 90 (天)	访问方法
GroomSummariesRetentionDays	DesktopGroupSummary、FailureLogSummary 和 LoadIndexSummary records。聚合数据 - 日粒度。	对于非 Platinum 用户为 7；对于 Platinum 为 90	Cmdlet (set/get-monitorconfiguration)
GroomHourlyRetentionDays	聚合数据 - 小时粒度	32 天	Monitor.Configuration Database Table。请参阅下面的注释。
GroomMinuteRetentionDays	聚合数据 - 分钟粒度	3 天	Monitor.Configuration Database Table。请参阅下面的注释。
GroomFailuresRetentionDays	MachineFailureLog 和 ConnectionFailureLog 记录	对于非 Platinum 用户为 7；对于 Platinum 为 90	Cmdlet (set/get-monitorconfiguration)
GroomLoadIndexesRetentionDays	LoadIndex 记录	对于非 Platinum 用户为 7；对于 Platinum 为 90	Cmdlet (set/get-monitorconfiguration)
GroomDeletedRetentionDays	LifecycleState 状态为“Deleted”的 Machine、Catalog、DesktopGroup 和 Hypervisor 条目。将删除任何相关的 Session、SessionDetail、Summary、Failure 或 LoadIndex 条目。	对于非 Platinum 用户为 7；对于 Platinum 为 90	Cmdlet (set/get-monitorconfiguration)
GroomMachineHotfixHistoryRetentionDays	应用至 VDA 和 Controller 计算机的修补程序	非 Platinum 和 Platinum 用户均为 90	Cmdlet (set/get-monitorconfiguration)

警告：修改 Monitor Service 数据库上的值需要重新启动此服务才能使新值生效。建议仅在 Citrix 技术支持人员的指导下修改 Monitor Service 数据库。

长期保留数据会对表格大小产生以下影响：

- **小时数据。**如果允许小时粒度的数据在数据库中最多保留两年，具有 1000 个交付组的站点将导致数据库按以下方式增长：
 $1000 \text{ 个交付组} \times 24 \text{ 小时/天} \times 365 \text{ 天/年} \times 2 \text{ 年} = 17,520,000 \text{ 行数据}$ 。聚合表中存在如此巨大的数据量对性能的影响是非常显著的。如果从此表格提取控制板数据，将需要使用大型数据库服务器。数据量过大会对性能造成巨大影响。
- **会话和事件数据。**这是指每次启动会话和建立连接/重新连接时收集的数据。对于大型站点（10 万个用户），此数据的增长速度非常快。例如，这些表格经过两年时间将大于 TB 级数据，这将要求使用企业级高端数据库。

使用 SSL 保护端点安全

May 04, 2016

本文档介绍了如何使用 SSL 为 Monitor Service OData API 端点提供安全保护。如果选择使用 SSL，则必须在站点内的所有 Delivery Controller 上配置 SSL。不能同时使用 SSL 和非 SSL。

要使用 SSL 为 Monitor Service 端点提供安全保护，必须执行以下配置。有些步骤需要在每个站点一次性完成，其余步骤则须在站点中托管 Monitor Service 的计算机上执行。各步骤的说明如下所述。

第 1 部分：在系统中注册证书

1. 使用可信证书管理器创建证书。证书必须与您希望用于 OData SSL 的计算机上的端口相关联。
2. 配置 Monitor Service，以使用此端口进行 SSL 通信。操作步骤取决于您的环境及其与证书结合使用的方式。下例说明了如何配置端口 449：

- 将证书与某个端口相关联：
`netsh http add sslcert ipport=0.0.0.0:449 certhash=97bb629e50d556c80528f4991721ad4f28fb74e9 appid='{00000000-0000-0000-0000-000000000000}'`
提示：在 PowerShell 命令窗口中，请务必用单引号引起 appId 中的 GUID（如下所示），否则命令将不起作用。请注意，在此示例中添加的换行符只是为了方便阅读。

第 2 部分：修改 Monitor Service 的配置设置

1. 在站点中的任意 Delivery Controller 上，运行以下 PowerShell 命令一次。此命令解除了 Monitor Service 在 Configuration Service 中的注册状态。
`asnp citrix.* $serviceGroup = get-configregisteredinstance -servicetype Monitor | Select -First 1 ServiceGroupUid remove-configserviceGroup -ServiceGroupUid $serviceGroup.ServiceGroupUic`
2. 在站点中的所有 Controller 上执行以下操作：
 - 使用 cmd 提示符查找已安装的 Citrix Monitor 目录（通常位于 C:\Program Files\Citrix\Monitor\Service 中）。在该目录下，运行：
`Citrix.Monitor.Exe -CONFIGUREFIREWALL -ODDataPort 449 -RequireODDataSsl`
 - 运行以下 PowerShell 命令：
`asnp citrix.* (if not already run within this window) get-MonitorServiceInstance | register-ConfigServiceInstance Get-ConfigRegisteredServiceInstance -ServiceType Config | Reset-MonitorServiceGroup`

示例

May 04, 2016

以下示例介绍如何使用 OData API 导出 Monitor Service 数据。

示例 1 - 原始 XML

1. 将每个数据集的 URL 置于正在使用 XenApp 或 XenDesktop 站点的相应管理权限运行的 Web 浏览器中。Citrix 建议使用装有 Advanced Rest Client 加载项的 Chrome 浏览器。
2. 查看来源。

示例 2 - PowerPivot with Excel

这些说明假定您已安装 Microsoft Excel 和 PowerPivot。

打开 Excel（使用 XenApp 或 XenDesktop 站点的相应管理权限运行）。

如果使用的是 Excel 2010：

1. 单击“PowerPivot”选项卡。
2. 单击 PowerPivot 窗口。
3. 单击功能区中的**从数据馈送**。
4. 选择友好的连接名称（例如：XenDesktop Monitoring Data）并输入数据馈送 URL：http://{dc-host}/Citrix/Monitor/OData/v2/Data（如果正在使用 SSL，则输入 https://{dc-host}/Citrix/Monitor/OData/v2/Data）。
5. 单击**下一步**。
6. 选择要导入到 Excel 的表并单击**完成**。这样便可检索到数据。
7. 现在，您可以使用 PowerPivot 查看和分析带有数据透视表和数据透视图的数据。有关详细信息，请参阅 Learning Center：<http://www.microsoft.com/en-us/bi/LearningCenter.aspx>

如果使用的是 Excel 2013：

1. 单击“数据”选项卡。
2. 依次选择“自其他来源”>“来自 OData 数据馈送”
3. 输入数据馈送 URL http://{dc-host}/Citrix/Monitor/OData/v1/Data（如果正在使用 SSL，则输入 https://{dc-host}/Citrix/Monitor/OData/v1/Data）并单击**下一步**。
4. 选择要导入到 Excel 的表并单击**下一步**。
5. 接受默认名称或自定义名称，然后单击**完成**。
6. 选择**仅连接或透视报表**。这样便可检索到数据。
7. 现在，您可以使用 PowerPivot 查看和分析带有数据透视表和数据透视图的数据。有关详细信息，请参阅 Learning Center：<http://www.microsoft.com/en-us/bi/LearningCenter.aspx>

示例 3 - LINQPad

这些说明假定您已安装 LINQPad。

1. 通过 XenApp 或 XenDesktop 站点的相应管理权限运行 LinqPad。
提示：最简便的方法是在 Delivery Controller 上下载、安装并运行。
2. 单击“添加连接”链接。
3. 选择 WCF Data Services 5.1 (OData 3) 并单击**下一步**。

4. 输入数据馈送 URL `http://{dc-host}/Citrix/Monitor/OData/v2/Data`（如果正在使用 SSL，则输入 `https://{dc-host}/Citrix/Monitor/OData/v2/Data`）。如有必要，请输入用户名和密码以访问 Delivery Controller。单击**确定**。
5. 现在您可以针对数据馈送运行 LINQ 查询，并根据需要导出数据。例如，在“目录”上单击鼠标右键并选择 **Catalogs.Take(100)**。此操作将返回数据库中前 100 个目录。选择“导出”>“导出到具有格式设置的 Excel”。

有关如何将该 API 用于 LINQPad 的更多实例，请参阅 <http://blogs.citrix.com/2014/01/14/creating-director-custom-reports-for-monitoring-xendesktop/>。

XenApp 和 XenDesktop SDK

May 04, 2016

XenApp 和 XenDesktop 提供基于多个 Microsoft Windows PowerShell 3.0 管理单元的 SDK，它允许您执行通过 Citrix Studio 控制台可以完成的任务，以及一些只使用 Studio 无法完成的任务。

自 7.5 版开始，XenApp 和 XenDesktop 共享统一的体系结构和管理：FlexCast Management Architecture。这意味着 XenApp 提供了多种之前仅在 XenDesktop 中提供的功能；因此，与常用功能相关的 SDK 元素同样适用于 XenApp 和 XenDesktop，即使命令本身仅引用 XenDesktop 也是如此。

- **全新的高级 SDK** — XenDesktop 7 提供了全新的高级 SDK，使您可以编写脚本、自动化站点创建以及轻松地进行快速维护。高级 SDK 为您免除了很多低级 SDK 的复杂性，例如，您只需运行两个 cmdlet 即可轻松创建一个新站点。
- **全新的低级 SDK** — 为全新的 XenDesktop 7 服务提供了单独的低级 SDK，包括为 Delegated Administration Service (DAS) 提供了增强的专用 SDK，此 SDK 之前在 XenDesktop 5 中是 Broker SDK 的一部分。此外，还提供了用于 Monitor Service、环境测试以及配置日志记录等新功能的 SDK。
- **Windows 服务器操作系统计算机目录和交付组** — 可以使用 XenDesktop 7 SDK 交付托管在服务器操作系统上的具有成本效益的应用程序和桌面。
- **桌面操作系统计算机应用程序** — 桌面操作系统计算机应用程序在 SDK 级别发生了显著变化。如果您具有用于在桌面操作系统上运行应用程序的现有脚本，则必须为 XenDesktop 7 更新这些脚本，因为几乎不进行向后兼容。
- **将设置应用于交付组中的计算机** — 在 XenDesktop 7 中，使用配置插槽，可以将设置应用于特定交付组中的计算机，而非站点中的所有计算机。这样，就可以配置应用于特定交付组的设置。系统提供了多个包含不同类型设置的预定义配置插槽，例如用于 Receiver 或 App-V 发布服务器位置的 StoreFront 地址设置。可以将插槽的一组设置仅应用于某个特定交付组，同时将同一插槽的另一组设置应用于另一交付组。可以使用与特定部署相对应的名称；例如，“销售部门策略”。
- **取代了目录类型** — 在 XenDesktop 7 中，已使用具有单个属性的目录取代了目录类型。但是，为了向后兼容，仍然可以使用采用目录类型的现有脚本，例如单一映像（池）和精简克隆（专用）等等，但是在内部这些目录类型会被转换为属性集。
警告：在可行的情况下，会维护与 XenDesktop 5 目录类型的向后兼容。但是，编写新脚本时，请勿使用目录类型；而是使用单个属性指定目录。
- **取代了桌面对象** — 在 XenDesktop 5 中，桌面对象是 Broker SDK 脚本中使用的 SDK 对象的主要类型之一。桌面对象同时描述了计算机和计算机上的会话。在 XenDesktop 7 中，此对象已被会话对象和计算机对象所取代，这两个对象均已扩展为担任桌面对象的工作。但是，为了向后兼容，仍然可以使用采用了桌面对象的现有脚本。
警告：在可行的情况下，会维护与 XenDesktop 5 的向后兼容。但是，编写新脚本时，请勿使用桌面对象；而是指定会话对象和计算机对象。

SDK 与 Studio 控制台在策略规则方面存在区别。在 SDK 中，授权和分配策略规则是独立的实体；但在控制台中，这些实体不可见，因为它们与交付组无缝合并在一起。此外，SDK 中访问策略规则的限制性更低。

SDK 由多个 PowerShell 管理单元组成，在安装 Controller 或 Studio 组件时，安装向导会自动安装这些管理单元。

访问并运行 cmdlet：

1. 在 PowerShell 3.0 中启动 shell。

要从控制台启动 shell，请单击 **Studio**，选择“PowerShell”选项卡，然后单击启动 **PowerShell**。

必须使用拥有 Citrix 管理员权限的身份运行 shell 或脚本。尽管在 Controller 上，本地管理员组的成员自动拥有完全管理权限以允许安装 XenDesktop，但 Citrix 建议，对于常规操作，应创建拥有相应权限的 Citrix 管理员，而不要使用本地管理员帐户。如果运行的是 Windows Server 2008，必须以 Citrix 管理员身份运行 shell 或脚本，而不要使用本地管理员组成员身份。

2. 要在脚本内使用 SDK cmdlet，应在 PowerShell 中设置执行策略。

有关 PowerShell 执行策略的详细信息，请参阅 Microsoft 文档。

3. 在 Windows PowerShell 控制台中使用 **Add -PSSnapin** 命令将需要的管理单元添加到 PowerShell 环境中。V1 和 V2 表示管理单元的版本（XenDesktop 5 管理单元为版本 1；XenDesktop 7 管理单元为版本 2）。例如，键入：

```
Add-PSSnapin Citrix.ADIdentity.Admin.V2
```

要导入所有 cmdlet，键入：

```
Add-PSSnapin Citrix.*.Admin.V*
```

导入后，可以访问 cmdlet 及其关联帮助。

有关典型用例的示例，请参阅 [SDK 入门](#)。

提示：有关 cmdlet 的所有帮助文本的完整列表，请参阅 [PowerShell cmdlet 帮助](#)。

Citrix Group Policy SDK 可用来显示并配置组策略设置和过滤器。它使用 PowerShell 提供程序创建与计算机和用户的设置及过滤器相对应的虚拟驱动器。提供程序以 New-PSDrive 扩展的形式显示。要使用 Group Policy SDK，必须安装 Studio 或 XenApp 和 XenDesktop SDK。

添加 Group Policy SDK

1. 要添加 Group Policy SDK，应键入：

```
Add-PSSnapin citrix.common.grouppolicy
```

2. 要访问帮助，应键入：

```
help New-PSDrive -path localgpo:/
```

使用 Group Policy SDK

1. 要创建虚拟驱动器并加载该驱动器及设置，应键入：

```
New-PSDrive [-PSProvider] CitrixGroupPolicy
```

— *-Controller*

```
New-PSDrive [-PSProvider] CitrixGroupPolicy
```

— *-Controller*

其中，

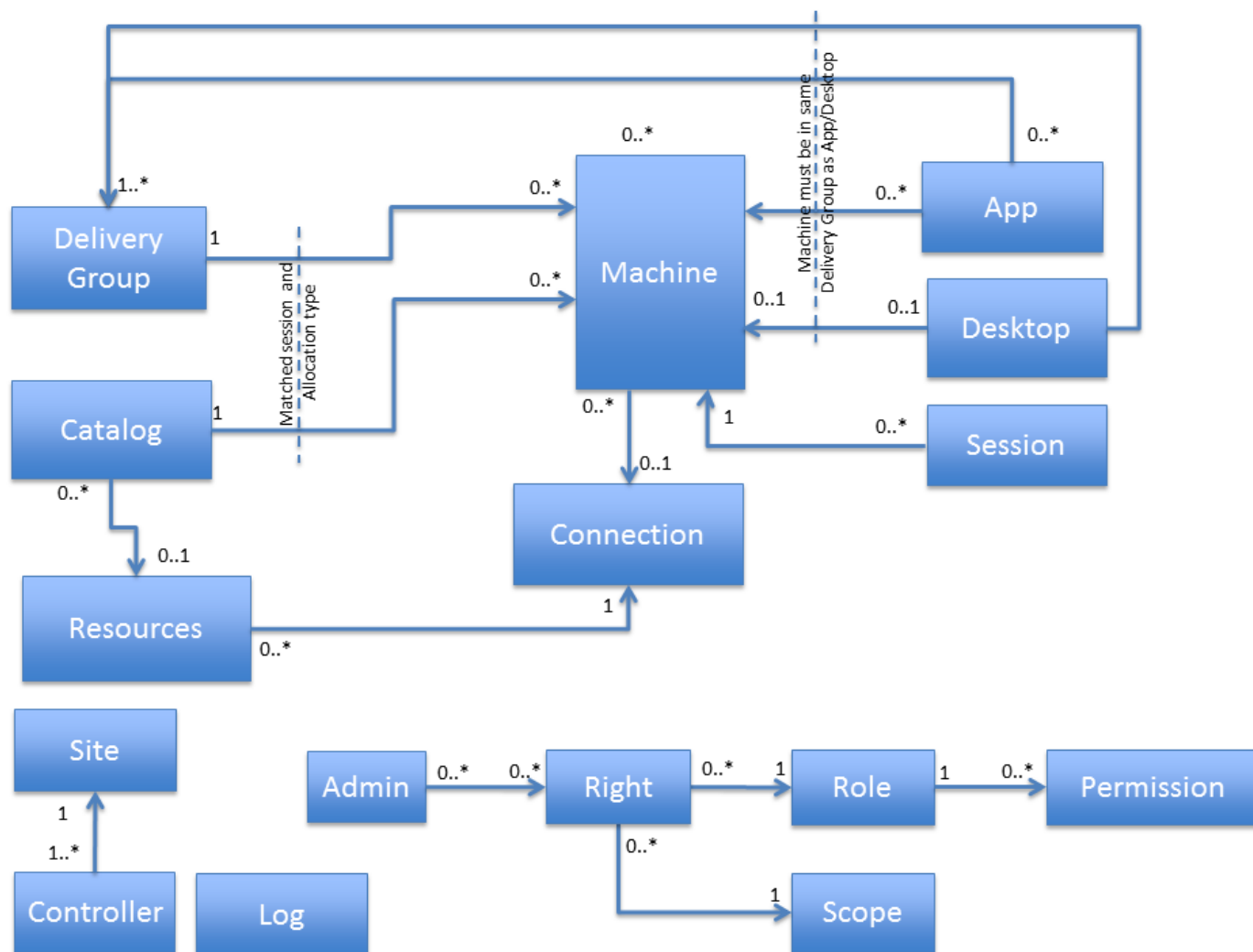
— *-Controller*

是站点中您要与之连接并从中加载设置的 Controller 的完全限定域名。

了解 XenDesktop 管理模式

May 04, 2016

XenDesktop 7 具有一种通过一组对象及其互连定义的管理模式，如下图所示。



以下部分将介绍各个对象、对象的职责、特征和基本属性。

连接

连接实体提供与虚拟化平台的管理点建立连接所需的详细信息。这些详细信息用于提供电源管理和所需的置备功能。连接定义：

- 连接的身份，包括名称和内部 ID 值 (GUID)。
- 类型 — XenServer、SCVMM 或 vCenter。
- 用户名 — 用于连接到虚拟机管理程序的帐户名称。
- 密码 — 用于连接到虚拟机管理程序的密码。
- 连接的状态以及是否可以与虚拟机管理程序通信。
- 虚拟机管理程序提出的电流警报。

密码设置后无法读回，并且以加密形式存储在数据库中。

在每种平台定义的模式中存在这些实体中的一个实体，但是，在运行时会与虚拟化平台建立多个连接，以服务于 XenDesktop 部署的要求。这些连接可供多个资源实体共享（请参阅下文以了解详细信息）。

注意：使用 XenServer 设置连接时，您指定主机或 XenServer 池；使用 Microsoft Hyper-V 或 VMware 设置连接时，您向管理服务器（SCVMM 或 vCenter）提供详细信息。此外，配置连接时，必须采用正确的格式指定用户名，具体取决于所使用的虚拟化解决方案。如果使用 XenServer 设置连接，指定纯文本用户名，如“username”，如果使用 VMware 或 Hyper-V 设置连接，则指定域用户名，例如“domain\username”。

资源

注意：此对象仅适用于使用 Machine Creation Services (MCS) 或 Provisioning Services (PVS) 创建新计算机。此设置定义虚拟化平台中供产品使用的一组资源。利用此设置，管理员可以限制部署以便仅使用平台所提供的资源的子集，并且可以针对不同的目的定义不同的资源集合。这样可以控制站点内置备的计算机的存储和网络使用情况。资源实体定义：

- 网络 — 向虚拟化平台置备新计算机时可以使用的网络（对于 PVS，是指流网络；只能选择一个或者使用第一个）。
- 存储 — 向虚拟化平台置备新计算机时使用的存储。

如果要使用置备机制，必须至少为每个连接对象定义一个资源设置。多个资源设置可以链接到同一个连接。

连接详细信息由连接实体提供。

注意：指定到 SCVMM 或 vCenter 的连接上的资源时，选择主机或群集，以及存储和网络。

管理员

此设置定义可以管理产品的人员。产品可以有多个管理员，每个管理员负责产品中的不同功能集合。管理员定义：

- 定义管理员的 AD 帐户，定义为管理员的名称或 SID 值。可以是单个用户或安全组。
- 管理员具有的权限（即，在哪些作用域上具有何种角色）。

计算机

计算机代表站点内可以用于向用户提供会话的虚拟机或物理计算机。计算机定义：

- 计算机的身份，如 SAM 名称、DNS 名称、主机（虚拟机管理程序）名称、IP 地址、SID、许可证 ID 等。
- 计算机的状态，包括电源状态、注册状态、个人虚拟磁盘状态、负载指数等。摘要状态值将这些状态汇总到一个状态值中。
- 关于计算机环境和配置的信息，如已安装操作系统和 Citrix 组件的版本号，包括计算机的“功能级别”。
- 有关计算机最近活动的信息，如最近一次取消注册的原因、最近一次在计算机上执行电源操作、最近一次连接故障等。
- 用于控制计算机的启用/禁用/终止行为的维护模式和“WindowsConnectionSetting”状态。
- 用于已分配桌面案例的计算机可见用户资源设置，如图标、已发布名称等。
- 对于多会话计算机（服务器操作系统计算机）：关于会话运行的汇总信息，如活动会话的数量、挂起会话的数量等。
- 与已分配计算机关联的“标记”值。
- 置备映像信息，如主映像的路径，用于创建计算机的置备方案、以及计算机是否具有在下次重新启动时挂起的更新映像。

大多数计算机值由 Broker Service SDK 展示，但是有些项目也通过其他服务展示，如 Machine Creation Service (MCS)。基于每个计算机显示的其他值继承自计算机所关联的目录或交付组（如有），或者来自用于创建计算机的方案。例如，通过多种因素（如物理还是虚拟、静态还是随机、用户更改的保留方式及其他）的组合在目录级别定义的计算机类型。对于单会话计算机（桌面操作系统计算机）：计算机上正在运行的会话（如有）的信息也与计算机关联。这包括连接的用户身份、会话状态、使用的协议等。如果将计算机定义为属于某个目录，计算机仅能存在于此模式下；不能存在于此概念之外。

目录

目录定义通常（并非始终）应该相同的一组计算机。单个部署中可以存在多个目录，从而可以针对不同的目的构建和存储不同的计算机集合。目录定义：

- 目录类型，通过以下属性值定义：
 1. 计算机的置备方法（MCS、PVS 或手动）
 2. 将目录中的计算机分配给用户的方式：永久分配给用户的静态方式或在用户每次请求资源时进行分配的随机方式
 3. 计算机是单会话（桌面操作系统计算机）还是多会话（服务器操作系统计算机）
 4. 计算机是物理计算机还是虚拟机
 5. 处理用户对计算机所做更改的方式，是在用户注销后丢弃、保留在计算机本地还是使用个人虚拟磁盘
- 目录是否用于 Remote PC 用户，如果是，哪些 Active Directory (AD) OU 与目录关联，以及哪些交付组与目录关联。
- 与 MCS 置备有关的详细信息（如果要使用 MCS 置备）：
 1. 计算机的主映像
 2. 内存大小和 CPU 数
 3. 个人虚拟磁盘大小、驱动器盘符和分配百分比
 4. AD 帐户命名方案和创建计算机的 OU，以及已经创建帐户的列表
- 与目录相关的任何 PVS 服务器的详细信息。
- 计算机的预期功能级别；不允许功能级别较低的计算机向站点注册。

在目录级别展示的信息还包括在目录中使用或提供的计算机的使用情况值。

交付组

交付组提供用于向最终用户提供桌面和/或应用程序的计算机集合的详细信息。多个交付组可以链接到同一个目录，从而可以根据不同用户集合的要求采用各种方式来分发目录中的计算机。交付组定义：

- 组中的计算机的分配类型，指示计算机是在多个用户之间共享（随机）还是永久分配给一个或多个用户（静态），以及计算机是单会话（桌面操作系统）计算机还是多会话（服务器操作系统）计算机。
- 组的交付类型，指示组是为用户仅提供应用程序、仅提供桌面，还是既提供应用程序又提供桌面。
- 控制组中的计算机电源管理的设置，包括：
 1. 将每周不同日期的哪个时间视为“高峰”时间
 2. 对于随机/未分配计算机，在每周不同日期的不同时间保持运行的计算机的数量，包括缓冲大小
 3. 针对上述设置评估每日时间的时区
 4. 是否以及如何对分配的计算机进行电源管理
 5. 是否以及如何在触发事件（如用户断开连接或注销）后关闭或挂起计算机
 6. 是否将计算机视为被其上运行的会话损坏，以及是否在每次使用后强制重新启动以将其返回到全新的已知状态
- 组中的桌面资源向最终用户的显示方式，包括使用的图标、颜色深度和名称。还会定义组中计算机的 ICA 连接所需的安全级别。对于桌面资源，允许每个用户从组并发访问的桌面数。
- 建立这些计算机对最终用户的可用性所使用的规则。规则不仅考虑用户的身份，还会考虑用户从何处采用何种方式进行连接、客户端设备的状态以及支持哪些远程处理协议。
- 是否启用组，包括维护模式的设置。
- 是否允许最终用户自行重置计算机（例如，使用 StoreFront）。
- 与随机/共享组关联的“标记”值。
- 组中的计算机的预期功能级别；不允许功能级别较低的计算机向站点注册。
- 组是否用于 Remote PC 用户，以及与组关联的 Remote PC 目录。
- 定期在特定时间和日期重新启动多会话（服务器操作系统）计算机的计划，以及用于控制如何执行重新启动的设置。

与交付组关联的设置还包括一些功能设置（如 Profile Management）和 Storefront URL 设置，分别定义为“计算机配置”对象并与一个或多个交付组关联。

应用程序

此对象提供要向最终用户提供的无缝（即浮动窗口，与桌面分开）应用程序的详细信息。通常，每个应用程序与单个交付组关联，但是如有需要，应用程序定义可以在多个组之间共享。应用程序既可以在远程计算机上运行并显示在本地客户端桌面上，

也可以安装并运行在本地客户端计算机上，而窗口覆盖在远程桌面上。应用程序定义：

- 应用程序的类型，“HostedOnDesktop”或“InstalledOnClient”。
- 对于 HostedOnDesktop 应用程序：
 1. 将在 VDA 计算机上运行的应用程序初始可执行文件的路径，以及应用程序启动时要应用的命令行参数（如有）。
 2. （可选）具有应用程序访问权限的特定用户集合，是具有交付组访问权限的用户的子集。
 3. 将应用到应用程序进程的应用程序特定设置，包括 CPU 优先级、应用程序是否应该等待创建代理打印机，等等。
- 对于 InstalledOnClient 应用程序：
 1. 指示应用程序的图标将从客户端设备获取的标志
 2. 指示要随为应用程序提供的参数一起采取额外安全措施的标志
- 应用程序资源向最终用户的显示方式，包括使用的图标、文件夹在客户端设备上的位置、名称以及是否在开始菜单和/或桌面上显示快捷方式。
- 应用程序的文件夹类型关联，用于将文件扩展名与应用程序关联。
- 所关联的交付组，以及用于在多个交付组之间进行选择的优先级值（可选）。
- 是否启用应用程序，以及是否对最终用户可见。

桌面

在 XenDesktop 7 中，会话对象和计算机对象取代了桌面对象（用于描述计算机及计算机上的会话），这二者已扩展到可以执行桌面对象的工作。

会话

此对象提供受站点控制的计算机上运行的 Windows 会话的详细信息。会话可以通过 XenDesktop 或 XenApp 启动的会话，也可以是通过其他方式创建的会话，如用户通过控制台或 RDP 直接登录到计算机上。

会话定义：

- 会话运行所在的计算机的身份，包括计算机 DNS 名称、IP 地址、NetBios 名称、SID 等。
- 运行会话的用户的身份，包括 SAM 名称、UPN、SID 等。
- 代理会话连接的用户的身份，包括 SAM 名称、SID 等。
- 用于连接到会话的端点客户端计算机的身份，包括设备名称、IP 地址、ID。
- 用于请求启动的计算机的身份，即执行启动的 Web 服务器。这包括名称和 IP 地址。
- 用于充当会话连接网关的计算机的身份，包括计算机 DNS 名称、IP 地址。
- 会话中的重要事件的详细信息，包括开始时间、最近连接到会话的时间、最近代理到会话的时间，等。
- 有关最近会话创建和会话连接的各种持续时间，包括代理会话的时间、创建会话的时间，等。
- 会话的当前状态，包括会话的总体状态、ICA 连接是否安全、正在使用的协议。
- 运行会话的计算机的当前状态，包括整体摘要状态、电源状态，等。
- 有关建立会话连接的详细信息，如会话是通过代理连接还是自主连接，有效的会话上下文“智能访问”标记。
- 会话是否“隐藏”。当用户启动应用程序或桌面时，如果会话遇到某些类型的问题，会话将变为隐藏。
- 在会话中执行的代理应用程序的列表。

会话 SDK 对象还提供来自相关计算机、交付组和目录 SDK 对象的信息。此信息包括身份信息、基本配置和状态信息。

Controller

此对象提供站点中的单个 Delivery Controller 计算机的详细信息。大部分数据是来自运行站点的动态状态数据，而不是配置设置。Controller 显示：

- 每个 Controller 上处于活动状态的“站点服务”。
- Controller 组件的版本。
- Controller 计算机的身份，如完全限定的 DNS 名称、SAM 名称、SID 等。

- Controller 计算机的操作系统类型和版本。
- 当前的 Controller 和服务状态，以及最近的活动时间。
- 向 Controller 注册的计算机的数量。
- 为了站点服务定位目的与 Controller 关联的虚拟机管理程序连接。

大多数 Controller 值由 Broker Service SDK 展示，但是有些项目通过其他服务展示，如 Machine Creation Service。

站点

站点是 XenDesktop 站点的顶层逻辑表示（从其中运行的配置服务角度）。站点包含许可信息、站点元数据和站点名称等。

XenDesktop 安装只有一个配置站点实例。站点对象具有以下属性：

- 名称
- 控制器 — 站点中的控制器列表
- 数据库 — 站点使用的数据库列表
- DefaultIconUid
- LicenseInformation
- 元数据

日志

此对象提供已收集的配置日志的详细信息，描述启用日志记录以来站点上的管理员活动。不记录管理员的读取操作，但是任何更改站点配置或状态的管理员操作均包含在日志中。可以在三个级别查看日志：高级日志、低级日志和操作详细信息。每个低级日志都是所记录的更大高级操作的一部分，而操作详细信息描述单个低级操作中的元素。日志项目显示：

- 执行操作的管理员的身份，包括执行操作所通过的计算机的 IP 地址。
- 操作的执行时间，包括开始时间和结束时间。
- 操作是成功还是失败及其原因。
- 操作的文本字符串说明，包括操作类型、源类型、目标类型等。
- 向操作提供的参数的详细信息。
- 在日志层次结构中的父/子关系。
- 所更改项目之前的值和之后的值（对于配置更改）。

有关所记录内容的详细信息，请参阅[配置日志记录](#)文档。

权限

此对象定义角色与角色权限所适用的作用域的组合。角色定义的权限可以由指定管理员执行，但是只能在与指定作用域直接或间接关联的对象上执行。权限定义：

- 允许其权限的角色。
- 角色权限所适用的作用域。

权限不是可单独操作的 SDK 对象，它们总是与特定的管理员对象关联。单个管理员可以具有多种权限，管理员的全部功能是其所有单个权限的总和。

作用域

此对象定义指定的对象分组，分组中的对象对以各个角色为基础进行控制的对象具有管理员权限。多种类型的 SDK 对象可以具有直接与其关联的作用域，如目录、交付组、连接、资源等。这些 SDK 对象具有列出与其关联的作用域列表的属性。其他 SDK 对象通过与具有直接关联作用域的对象进行二级关联而获得管理员权限。例如，计算机对象从其所属的目录和交付组继承

作用域关联；会话对象从其运行所在的计算机继承作用域关联。有些作用域是预定义的（在实际中当前仅内置了“全部”作用域），但是您可以创建其他作用域以指定适合特定部署的分组定义。

作用域定义：

- 作用域的身份，包括名称和内部 ID 值 (GUID)。
- 作用域是否为内置作用域。

角色

定义管理员可以执行的一组权限。角色始终被授给具有关联作用域的管理员；它们并不向自己提供权限，尽管有些角色可能具有常规权限，这些权限适用于不具有关联作用域的对象（无作用域对象）。有些角色是预定义角色，但是您可以创建其他自定义角色以指定适合特定部署的权限集合。角色定义：

- 角色的身份，包括名称和内部 ID 值 (GUID)。
- 角色是否为内置角色。
- 组成角色的权限集。

内置角色包括：

- 完全权限管理员 — 可以执行所有任务和操作。
- 只读权限管理员 — 可以查看指定作用域内的所有对象以及全局信息，但无法更改任何内容。
- 计算机目录管理员 — 可以创建和管理计算机目录以及置备计算机。
- 交付组管理员 — 可以交付应用程序、桌面和计算机；还可以管理关联的会话。允许创建和管理交付组 and 应用程序。
- 技术支持管理员 — 可以查看交付组，并可以管理与这些组关联的会话和计算机。允许查看最终用户资源和有限的状态更改操作，以便为最终用户排除故障，但是不允许执行大多数配置更改。
- 主机管理员 — 可以管理主机连接及其关联的资源设置。

权限

定义当角色包含权限时所允许执行的单个控制台级别任务或操作。每个权限可以允许多个低级别的 SDK 操作 (cmdlet)，特定的低级别 SDK 操作可以通过任意数量的相关权限授权。权限定义：

- 权限的身份，包括名称和内部 ID 值 (GUID)。
- 权限的权限组成员身份。权限组将与特定功能区域相关的权限集合在一起。
- 权限涵盖低级别 SDK 操作集。

SDK 入门

May 04, 2016

要创建脚本，请执行以下步骤：

1. 使用 Citrix Studio 执行要为其编制脚本的操作，例如，创建一组 Machine Creation Services 计算机的目录。
2. 收集 Studio 为执行此任务所进行的 SDK 操作的日志。
3. 查看脚本以了解每个部分的作用。这将有助于您自定义自己的脚本。有关详细信息，请参阅示例用例，其中详细解释了脚本所进行的操作。
4. 转换并修改 Studio 脚本段，将其转变成更适用的脚本。为此，您需要：
 - 使用变量。一些 cmdlet 带有参数，如 TaskId。但是，由于 Studio 使用早期 cmdlet 的结果对象的值，可能无法清楚了解这些参数中使用的值来自何处。
 - 删除任何不需要的命令。
 - 将一些步骤添加到循环中，以方便进行控制。例如，将计算机创建过程添加到一个循环中，这样便可以控制要创建的计算机数。

示例

注意：创建脚本时，为确保始终获得最新的增强功能和修复，Citrix 建议您按照本主题中所述的步骤进行操作，而不要复制粘贴示例脚本。

示例	说明
创建目录	脚本：创建一组 Machine Creation Services (MCS) 计算机的目录
示例：创建和配置主机	脚本：创建和配置主机
示例：创建 PvD 桌面	脚本：创建包含 Personal vDisk (PvD) 桌面的交付组
示例：获取负载平衡信息	显示服务器操作系统计算机的负载指数值

示例：创建目录

May 04, 2016

下例显示了如何为一组 Machine Creation Services (MCS) 计算机创建目录。

在开始之前，请务必按照 [SDK 入门](#) 中详述的步骤操作。本文档介绍了如何使用 Studio 执行要编写脚本的操作（在本例中，是为一组 Machine Creation Services 计算机创建目录），以及如何收集 Studio 为执行此任务所进行的 SDK 操作的日志。之后，可以自定义此输出，以生成自动完成目录创建脚本。

注意：为确保始终获得最新的增强功能和修复，Citrix 建议您按照本文档中所述的步骤进行操作，而不要复制粘贴示例脚本。为增强可读性，在此脚本中添加了行号和换行符。

了解脚本

以下部分介绍由 Studio 生成的脚本各部分的作用。这将有助于您自定义自己的脚本。为了增强可读性，添加了行号。

1. Start-LogHighLevelOperation -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -Source 'Studio' -StartTime 29/05/2013 14:43:08 -Text 'Create Machine Catalog "ExampleMachines"'

2. New-BrokerCatalog -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -AllocationType 'Permanent' -Description 'Example Machines' -IsRemotePC \$False -LoggingId f39a2792-064a-43eb-97c7-397c

3. New-AcctIdentityPool -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -AllowUnicode -Domain 'dumdev.internal.citrix.com' -IdentityPoolName 'ExampleMachines' -LoggingId f39a2792-064a-43eb-9

创建标识池。此操作定义用于创建 AD 计算机帐户的机制。标识池将用作即将创建的计算机而创建的 AD 帐户的容器。

4. Set-BrokerCatalogMetadata -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -CatalogId 1 -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46 -Name 'Citrix_DesktopStudio_IdentityPoolUid' -Value '1'

在 Broker 目录中，利用标识池的详细信息设置元数据。此步骤不是必需的步骤。

5. Test-ProvSchemeNameAvailable -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -ProvisioningSchemeName @('ExampleMachines')

检查请求的名称是否可用。此步骤不是必需的步骤。

6. New-ProvScheme -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -CleanOnBoot -HostingUnitName 'SharedNFS' -IdentityPoolName 'ExampleMachines' -LoggingId f39a2792-064a-43eb-97c7-397c7c7c7c7c

创建置备方案对象。这是要创建的计算机的模板。其中指定了虚拟机管理程序、网络、存储、内存和要使用的 CPU 数等。它采用系统中已设置的参数，例如 HostingUnit 名称和要用于待创建计算机的 VM 快照的路径。此命令“完整”复制要使用的 VM 快照，因此，此过程可能需要一段时间才能完成。

在此示例中，Studio 脚本指定了此命令中的 `-RunAsynchronous` 标志。这意味着，此命令会在运行结束之前将控制权返回给管理员。因此，必须等命令运行结束再执行任何需要此命令完成才能执行的操作。如果未指定此标志，此命令将以内联方式同步运行，并且直到命令运行完成（成功或失败）后，才会返回控制权。可以使用 `Get-ProvTask cmdlet` 查看异步任务的状态。提供启动任务的操作时返回的任务 ID。在本例中为 `New-ProvScheme cmdlet`。

7. Set-BrokerCatalog -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46 -Name 'ExampleMachines' -ProvisioningSchemeId 76125e3a-9001-4993-8000-000000000000

执行上述步骤中创建的置备方案的唯一 ID 更新 BrokerCatalog。

8. Add-ProvSchemeControllerAddress -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -ControllerAddress @('DDC.dumdev.internal.citrix.com') -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46 -

在置备方案对象中，添加一组控制器地址。这是一组地址列表，在部署时，创建的计算机可以使用这些地址在控制器 (Broker) 中注册。可以通过多种方式提供计算机的注册地址。但是，如果管理员要使用 VDA 安装程序中的“允许 Machine Creation Service 提供此信息”，则必须提供此信息。对此列表所做的变更仅影响更改后创建的计算机，不会影响现有计算机。

9. Get-AcctADAccount -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -IdentityPoolUId b99aee6d-8772-4dbc-978b-8eb9a26e2407 -Lock \$False -MaxRecordCount 2147483647 -State 'Available'

Studio 可以从标识池获取可用计算机标识的列表。这样一来，如果存在过去已创建但尚未使用的现有帐户，可以使用这些帐户而不必创建新帐户。请注意，在脚本中这不是必需的，因为只要脚本运行时所在的上下文具有创建新帐户的权限，即可采用创建新帐户的方式。但是，如果脚本无权创建新帐户，请将脚本改为使用可用帐户（运行脚本前，需要执行单独的过程，以便为标识池提供一组帐户）。

10. New-AcctADAccount -AdminAddress 'ddc.dumov.internal.citrix.com:80' -Count 2 -IdentityPoolUid b99aee6d-8772-4dbc-978b-8eb9a26e2407 -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46

在 Active Directory 中，创建所需的 AD 计算机帐户。此脚本将创建一个帐户，但如有必要，也可以使用命令中的“Count”参数创建更多帐户。创建的帐户将存储在通过上述步骤创建的置备方案中定义的 OU 中。

11. New-ProvVM -ADAccountName @"(DUMDEV\Example-0001\$';DUMDEV\Example-0002\$') -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46

根据上述步骤中创建的置备方案中的模板定义，创建虚拟机。此过程可能需要一段时间才能完成。

12. Lock-ProvVM -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46 -ProvisioningSchemeName 'ExampleMachines' -Tag 'Brokered' -VMID '@(071

锁定已置备的虚拟机，并防止意外修改虚拟机。SDK 用户可以以此来指示虚拟机正在使用以及锁定原因。此脚本通过“已设置代理”标志锁定 VM。该标志指示虚拟机已创建并添加到 Broker 目录，必须先将其从目录中移除，才能进行删除。可以根据需要设置标记名称。

13. New-BrokerMachine -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -CatalogUid 1 -HostedMachineId '0710bb77-d01f-d006-4d67-5472e5cd349f' -HypervisorConnectionUid 1 -LoggingId f39a279

创建 Broker 计算机对象。这些对象存储在目录中，并将随目录一起加入已置备的计算机。

14. Start-BrokerMachinePvdImagePrepare -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -InputObject @2) -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46
请求 Broker Service 启动 Personal vDisk 的准备操作。这是计算机初始化 Personal vDisk 存储所必需的操作。

15. Stop-LogHighLevelOperation -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -HighLevelOperationId f39a2792-064a-43eb-97c7-397cc1238e46 -lsSuccessful \$true
停止第一步中开始的记录操作，并指示操作成功。

自定义脚本

以下部分介绍如何将 Studio 输出转换并修改成更适用的脚本。除使用变量和删除不需要的命令外，还介绍了如何通过计算机创建循环过程来控制所创建计算机的数量。为了增强可读性，添加了行号。

1. CmdletBinding() param ([Parameter(Mandatory=\$true)] [string] \$hostingUnitPath, [Parameter(Mandatory=\$true)] [string] \$catalogName, [string] \$catalogDescription, [Parameter(Mandatory=\$true)] [int]

示例：创建和配置主机

May 04, 2016

以下示例介绍主机的创建和配置方法。

开始之前，请务必按照 [SDK 入门](#) 中详细介绍的步骤进行操作，该文档说明了如何使用 Studio 执行要编写脚本的操作（在此示例中，操作为创建主机）以及收集 Studio 为执行任务而做出的 SDK 操作的日志。之后，可以自定义此输出，以生成用于自动完成主机创建的脚本。

注意：为确保始终获得最新的增强功能和修复，Citrix 建议您按照本文档中所述的步骤进行操作，而不要复制粘贴示例脚本。为增强可读性，在此脚本中添加了行号和换行符。

了解脚本

以下部分介绍由 Studio 生成的脚本各部分的作用。这将有助于您自定义自己的脚本。为了增强可读性，添加了行号。

1. Get-LogSite -AdminAddress 'mycontroller.example.com:80'

查询配置日志记录服务，以检索有关站点配置的信息。

2. Start-LogHighLevelOperation -AdminAddress 'mycontroller.example.com:80' -Source 'Studio' -StartTime 14/08/2013 14:30:28 -Text 'Create Connection "Example XenServer"'

通过配置日志记录操作启动高级别日志记录操作（其中将包含其余命令）。将提供的日志 ID 返回给后续操作。

3. Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'

设置将由配置 cmdlet 使用的主机服务的位置。由于主机服务提供 PowerShell 提供程序，而且并非所有 cmdlet 都能取得服务的地址，所以此 cmdlet 将设置默认位置。

4. New-Item -ConnectionType 'XenServer' -HypervisorAddress @('http://xenhost1.example.com') -LoggingId e355ce51-8cbb-400a-ae81-1fdc567239cb -Path @('XDHyp:\Connections\Example XenServer')
与 XenServer (xenhost1.example.com) 建立连接。这并非永久连接，仅可供此 PowerShell 运行空间使用。

5. Stop-LogHighLevelOperation -AdminAddress 'mycontroller.example.com:80' -EndTime 14/08/2013 14:30:29 -HighLevelOperationId 'e355ce51-8cbb-400a-ae81-1fdc567239cb' -IsSuccessful \$True
停止先前开始的记录操作，并指示操作成功。

6. Get-LogSite -AdminAddress 'mycontroller.example.com:80'

查询配置日志记录服务，以检索有关站点配置的信息。

7. Start-LogHighLevelOperation -AdminAddress 'mycontroller.example.com:80' -Source 'Studio' -StartTime 14/08/2013 14:30:30 -Text 'Update Connection "Example XenServer"'

启动新的高级别日志记录操作。

8. Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'

重新设置主机服务地址详细信息（请注意，下面优化的脚本中删除了此重复操作）。

9. Set-Item -HypervisorAddress @('http://xenhost1.example.com','http://xenhost2.example.com') -LoggingId 44e15629-6906-4840-a36c-984aaf67be6d -PassThru -Path @('XDHyp:\Connections\Example XenServer')
更新第 4 步中创建的连接。由于池中存在多个 XenServer，因此它将提供所有地址来实现高可用性。

10. Stop-LogHighLevelOperation -AdminAddress 'mycontroller.example.com:80' -EndTime 14/08/2013 14:30:31 -HighLevelOperationId '44e15629-6906-4840-a36c-984aaf67be6d' -IsSuccessful \$True
停止第 7 步中开始的日志记录操作。

11. Get-LogSite -AdminAddress 'mycontroller.example.com:80'

查询配置日志记录服务，以检索有关站点配置的信息。

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'
开始执行新的日志记录操作。

13. Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'

重新设置主机服务地址详细信息。

14. Get-ChildItem -Path @('XDHyp:\Connections')

获取主机连接的内容以填充向导对话框。

15. Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'

重新设置主机服务地址详细信息。

16. Remove-Item -LoggingId 76caa3f4-df93-4cb2-b78d-6a8824766314 -Path @('XDHyp:\Connections\Example XenServer')

删除向导中创建的临时连接。

17. Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'

重新设置主机服务地址详细信息。

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')
以持久连接的形式重新创建连接，连接将写入数据库并可供其他 PowerShell 运行空间使用。

19. New-BrokerHypervisorConnection -AdminAddress 'mycontroller.example.com:80' -HypHypervisorConnectionUid a14096ba-5074-44ff-b596-371e345c0449 -LoggingId 76caa3f4-df93-4cb2-b78d-6a8824766314
将主机连接添加到 Broker 服务。

20. Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'

重新设置主机服务地址详细信息。

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')
使用第 14 步中收集的信息创建托管单元（在 Studio 中称为“资源”）。

22. Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'

重新设置主机服务地址详细信息。

23. Get-Item -Path @('XDHyp:\Connections\Example XenServer')

检索新创建的对象。

Stop-LogHighLevelOperation -AdminAddress 'mycontroller.example.com:80' -EndTime 14/08/2013 14:31:07 -HighLevelOperationId '76caa3f4-df93-4cb2-b78d-6a8824766314' -IsSuccessful \$True
停止先前开始的记录操作，并指示操作成功。

自定义脚本

以下部分介绍如何将 Studio 输出转换并修改成更适用的脚本。以下脚本经过简化，与上面的 Studio 脚本不同，在向导中收集信息期间，它不会创建临时主机连接，而是创建持久连接。之后，将从此连接内查询信息来创建托管单元（资源）。请注意，LoggingId 和 HypHyperConnectionUid 详细信息不同。

为方便阅读，添加了行号；每个编号项是一个 PowerShell 命令。

```
1. Start-LogHighLevelOperation -AdminAddress 'mycontroller.example.com:80' -Source 'Studio' -Text 'Create Connection "Example XenServer"' 2. Set-HypAdminConnection -AdminAddress 'mycontrolle
```

示例：创建 PvD 桌面

May 04, 2016

本文档提供了用于创建包含 Personal vDisk (PvD) 桌面的交付组的脚本示例。

开始之前，请务必按照 [SDK 入门](#) 中详细介绍的步骤进行操作，该文档说明了如何使用 Studio 执行要编写脚本的操作以及收集 Studio 为执行任务而做出的 SDK 操作的日志。之后，可以自定义此输出，以生成自动完成任务的脚本。

注意：为确保始终获得最新的增强功能和修复，Citrix 建议您按照本文中所述的步骤进行操作，而不要复制粘贴示例脚本。

了解脚本

以下部分介绍由 Studio 生成的脚本各部分的作用。这将有助于您自定义自己的脚本。为增强可读性，在此脚本中添加了行号和换行符。

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''
开始执行记录的操作并返回日志 ID，用于将后续操作与大型任务相关联。
2. New-BrokerDesktopGroup -AdminAddress 'test-ddc.mydomain.com:80' -ColorDepth 'TwentyFourBit' -DeliveryType 'DesktopsOnly' -DesktopKind 'Private' -InMaintenanceMode \$False -IsRemotePC \$F
通过 Studio 向导收集的选项创建一个新交付组。
3. Add-BrokerMachinesToDesktopGroup -AdminAddress 'test-ddc.mydomain.com:80' -Catalog 'win7-pvd' -Count 2 -DesktopGroup 'Win7 PvD Desktops' -LoggingId 846f2d42-a994-4bce-ab58-be05c8d73b99
将从指定目录请求的计算机数添加到新交付组。
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.mydo
不需要执行上述命令，Studio 正在验证用户。
5. Test-BrokerAssignmentPolicyRuleNameAvailable -AdminAddress 'test-ddc.mydomain.com:80' -Name @('Win7 PvD Desktops')
Studio 检查策略分配名称是否可用。
6. New-BrokerAssignmentPolicyRule -AdminAddress 'test-ddc.mydomain.com:80' -DesktopGroupUid 41 -Enabled \$True -IncludedUserFilterEnabled \$False -LoggingId 846f2d42-a994-4bce-ab58-be05c
为交付组创建新策略分配规则。未在此处指定任何用户，因此，所有控制操作都通过访问策略规则实现。
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.mydo
不需要执行上述命令，Studio 正在执行进一步的检查。
8. Test-BrokerAccessPolicyRuleNameAvailable -AdminAddress 'test-ddc.mydomain.com:80' -Name @('Win7 PvD Desktops_Direct')
Studio 测试访问策略规则是否可用。
9. New-BrokerAccessPolicyRule -AdminAddress 'test-ddc.mydomain.com:80' -AllowedConnections 'NotViaAG' -AllowedProtocols @('HDX','RDP') -AllowRestart \$True -DesktopGroupUid 41 -Enabled \$1
为新桌面创建访问策略规则以实现非 NetScaler Gateway 连接。
10. Test-BrokerAccessPolicyRuleNameAvailable -AdminAddress 'test-ddc.mydomain.com:80' -Name @('Win7 PvD Desktops_AG') New-BrokerAccessPolicyRule -AdminAddress 'test-ddc.mydomain.cor
Studio 重复此过程以实现 NetScaler Gateway 连接。
11. Test-BrokerPowerTimeSchemeNameAvailable -AdminAddress 'test-ddc.mydomain.com:80' -Name @('Win7 PvD Desktops_Weekdays') New-BrokerPowerTimeScheme -AdminAddress 'test-ddc.myc
Studio 检查（可选）工作日和周末电源方案的名称是否可用，然后添加这些名称。
12. Stop-LogHighLevelOperation -AdminAddress 'test-ddc.mydomain.com:80' -EndTime 31/07/2013 10:09:05 -HighLevelOperationId '846f2d42-a994-4bce-ab58-be05c8d73b99' -IsSuccessful \$True
停止第 1 步中开始的所记录操作，并指示操作成功。

自定义脚本

本部分介绍如何将 Studio 输出转换并修改成更适用的脚本。

此脚本将创建一个包含 PvD 桌面的交付组。在参数中指定的目录必须已存在且能适当地填充（填充静态和 PvD 类型的磁盘）。此脚本设计为由以 Citrix 管理员身份登录的用户从 Powershell 命令行运行。不针对权限执行任何检查；如果用户没有相应的权限，此脚本将失败。

<# 示例用法： .\CreatePvDGroup.ps1 -GroupName "Win7 PvD Desktops" -SrcCatalog "win7-pvd" -NumDesktops 2 -Users @('mydomain\user1','mydomain\user2') -#> Param([Parameter(Mandatory
下表说明了此脚本中使用的参数。

参数	说明
GroupName	新 PvD 桌面组的名称
SrcCatalog	用于创建 PvD 桌面的目录的名称。通过指定静态分配类型来创建目录。计算机还必须具有 PvD 磁盘。
NumDesktops	添加到 PvD 桌面组的计算机数。如果可用计算机数不足，则将添加尽可能多的计算机。
用户	哪些用户可以访问桌面组。这是用户或组的列表，例如 @('mydomain\Domain Users') 或 @('mydomain\user1','mydomain\user2')

Set-HypAdminConnection -AdminAddress \$adminAddress
指定要使用的虚拟机管理程序的管理连接。对某些命令而言，不再需要使用 -AdminAddress。

\$peakPoolSize = 2 \$weekendPoolSizeByHour = new-object int[] 24 \$weekdayPoolSizeByHour = new-object int[] 24 9..17 | %{ \$weekdayPoolSizeByHour[\$_] = \$peakPoolSize } \$peakHours = (0..23 | %{
这将创建 24 元素阵列，其中每个条目中包含一个 1 或 0。使用这些阵列可为交付组指定电源计划的高峰时段。工作日的元素 9 到 17（从 09:00 到 17:00 的时段）设置为 1，其他则保留为 0。如果可用，
则会在高峰时间启动两个未分配的计算机。

\$logId = Start-LogHighLevelOperation -Text "Create PvD desktop group" -Source "Create PvD Desktop Group Script"
开始执行新的记录的操作。此操作将返回一个日志 ID，该 ID 将传递到后续操作，以将其与创建组任务相关联。

```
$grp = New-BrokerDesktopGroup -DesktopKind 'Private' -DeliveryType 'DesktopsOnly' -LoggingId $logId -Name $GroupName -PublishedName $GroupName -SessionSupport 'SingleSession'
```

创建新交付组，用于交付专用桌面。使用的目录必须已填充适合的计算机（永久性具有 PVD 磁盘）。PublishedName 是向最终用户显示的名称；以下各项使用的名称与组名称相同。

```
New-BrokerAssignmentPolicyRule -DesktopGroupUid $grp.Uid -IncludedUserFilterEnabled $False -LoggingId $logId -MaxDesktops 1 -Name ($GroupName + '_AssignRule') | Out-Null
```

已分配的桌面需要具有分配策略。禁用用户过滤器以便完全通过访问策略规则对访问进行控制。

```
New-BrokerAccessPolicyRule -AllowedConnections 'NotViaAG' -AllowedProtocols @('HDX','RDP') -AllowRestart $True -DesktopGroupUid $grp.Uid -IncludedSmartAccessFilterEnabled $True -li
```

指定任何访问限制：允许使用 NetScaler Gateway 以及 HDX 和 RDP 协议进行直接访问。如有需要，用户可以请求重新启动桌面。

```
New-BrokerPowerTimeScheme -DaysOfWeek 'Weekdays' -DesktopGroupUid $grp.Uid -DisplayName 'Weekdays' -LoggingId $logId -Name ($GroupName + '_Weekdays') -PeakHours $peakHr
```

可选：指定电源计划。

```
Stop-LogHighLevelOperation -HighLevelOperationId $logId -IsSuccessful $True
```

停止配置日志记录并指示是否成功。

示例：获取负载平衡信息

May 04, 2016

可以使用服务器操作系统计算机向多个用户交付托管在服务器操作系统上的经济高效的应用程序和桌面。

要平衡部署中服务器操作系统计算机的负载，可以使用 Citrix 策略。许多负载平衡策略设置都可以用来在交付 Windows 服务器操作系统计算机的服务器之间启用和配置负载管理。有关详细信息，请参阅负载管理策略设置参考文档。可以通过 Studio 或 Windows 中的组策略管理控制台使用策略；有关详细信息，请参阅“策略”文档。

要查看负载，可以使用 Citrix Director 或 Studio 控制台或者 PowerShell SDK。下例显示了如何使用 PowerShell SDK 显示负载。

注意：如果使用过早期版本的 XenDesktop，您可能很熟悉 `qfarm /load` 命令。此工具不再可用，但可以使用 PowerShell 显示如下例所示的类似输出。

示例：获取负载指数值

要显示计算机列表、计算/测量的负载指数值以及计算机上运行的会话数，请执行以下操作：

1. 在 PowerShell 中启动 shell。有关详细信息，请参阅 [XenApp 和 XenDesktop SDK](#)。
2. 类型：

```
Get-BrokerMachine -SessionSupport MultiSession -Property 'DnsName','LoadIndex','SessionCount'
```

注意：负载指数值最大可以是 10000。负载指数值表示从已配置的源计算得出的 VDA 计算机负载，例如会话数量。值 10000 表示 VDA 计算机满载；Broker 不再向该计算机发送其他用户会话。

有关详细信息和示例，请参阅 `get-brokmachine cmdlet` 的 `cmdlet 帮助` 以及 `about_broker_filtering-xd7.html` 等“关于”主题。请参阅 [PowerShell cmdlet 帮助](#)。

PowerShell cmdlet 帮助

May 04, 2016

本部分提供了所有 cmdlet 的 PowerShell 帮助文本。

- 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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Gets the AD accounts stored in the AD Identity Service.

Syntax

```
Get-AcctADAccount [-IdentityPoolName <String>] [-ADAccountSid <String>] [-Domain <String>] [-State <ADIdentityState>] [-Lock <Boolean>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-AcctADAccount [-IdentityPoolUid <Guid>] [-ADAccountSid <String>] [-Domain <String>] [-State <ADIdentityState>] [-Lock <Boolean>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Provides the ability to locate the AD accounts stored within the AD Identity Service and view the state of the accounts.

Related topics

[New-AcctADAccount](#)

[Add-AcctADAccount](#)

[Remove-AcctADAccount](#)

[Unlock-AcctADAccount](#)

[Update-AcctADAccount](#)

[Repair-AcctADAccount](#)

Parameters

-ADAccountSid<String>

The AD Account SID of the account.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Domain<String>

The domain of the account (this is in dns format).

Required?	false

Default Value	
Accept Pipeline Input?	false

-State<ADIdentityState>

The current state of the identity stored in the AD Identity Service for the AD account.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Lock<Boolean>

Indicates if the account is locked in the AD Identity Service.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

See about_Acct_Filtering for details.

Required?	false
Default Value	false
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

See about_Acct_Filtering for details.

Required?	false

Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

See about_Acct_Filtering for details.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

See about_Acct_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Filter<String>

See about_Acct_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	LocalHost. Once a value is provided by any cmdlet, this value becomes the default.

Accept Pipeline Input?	false

-IdentityPoolName<String>

The name of the identity pool to which the account is registered.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-IdentityPoolUid<Guid>

The unique identifier for the identity pool that the account is registered to.

Required?	false
Default Value	
Accept Pipeline Input?	false

Return Values

Citrix.AdIdentity.Sdk.IdentityInPool

The Get-AcctADAccount returns an object that contains the following parameters

ADAccountSid <string>

The AD account SID for the retrieved account.

ADAccountName <string>

The AD account name for the retrieved account.

Domain <string>

The domain for the imported account.

State <Citrix.XDInterServiceTypes.ADIdentityState>

The state for the account. This can be;

Available

The account is not used.

InUse

The account is in use.

Error

The account is in error (i.e. the account is locked or disabled in AD).

Tainted

The account is no longer used, but the password is no longer known.

Lock <Boolean>

The account is locked (in the database not in AD).

IdentityPoolName <System.String>

The name of the containing identity pool.

IdentityPoolUid <System.Guid>

The GUID identifying the containing identity pool.

Notes

In the case of failure the following errors can result.

Error Codes

PartialData

Only a subset of the available data was returned.

CouldNotQueryDatabase

The query required to get the database was not defined.

PermissionDenied

The user does not have administrative rights to perform this operation.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

CommunicationError

An error occurred while communicating with the service.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

C:\>Get-AcctADAccount

Return all the AD accounts that are registered in the AD Identity Service.

----- EXAMPLE 2 -----

C:\>Get-AcctADAccount -IdentityPoolName MyPool -Lock \$false

Return all the AD accounts that are registered in the AD Identity Service in the identity pool called "MyPool" that are also locked.

----- EXAMPLE 3 -----

C:\>Get-AcctADAccount -Filter {IdentityPoolName -Like "p*" -or IdentityPoolName -eq "MyPool"}

Return all the AD accounts that are registered in the AD Identity Service in the identity pool called "MyPool" or in an identity pool that has a name that starts with a 'p'. For full details of the advanced filtering aspects of this command see about_Acct_Filtering.

Get-AcctDBConnection

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Gets the details of the scoped objects for the AdIdentity Service.

Syntax

```
Get-AcctScopedObject [-ScopeId <Guid>] [-ScopeName <String>] [-ObjectType <ScopedObjectType>] [-ObjectId <String>] [-ObjectName <String>] [-Description <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Returns a list of directly scoped objects including the names and identifiers of both the scope and object as well as the object description for display purposes.

There will be at least one result for every directly scoped object. When an object is associated with multiple scopes the output contains one result per scope duplicating the object details.

No records are returned for the All scope, though if an object is not in any scope a result with a null ScopeId and ScopeName will be returned.

Related topics

Parameters

-ScopeId<Guid>

Gets scoped object entries for the given scope identifier.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-ScopeName<String>

Gets scoped object entries with the given scope name.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-ObjectType<ScopedObjectType>

Gets scoped object entries for objects of the given type.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-ObjectId<String>

Gets scoped object entries for objects with the specified object identifier.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-ObjectName<String>

Gets scoped object entries for objects with the specified object identifier.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Description<String>

Gets scoped object entries for objects with the specified description.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about_Acct_Filtering for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell-style filter expression. See about_Acct_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

Citrix.AdIdentity.Sdk.ScopedObject

The Get-AcctScopedObject command returns an object containing the following properties:

ScopeId <Guid?>

Specifies the unique identifier of the scope.

ScopeName <String>

Specifies the display name of the scope.

ObjectType <ScopedObjectType>

Type of the object this entry relates to.

ObjectId <String>

Unique identifier of the object.

ObjectName <String>

Display name of the object

Description <String>

Description of the object (possibly \$null if the object type does not have a description).

Notes

If the command fails, the following errors can be returned.

Error Codes

UnknownObject

One of the specified objects was not found.

PartialData

Only a subset of the available data was returned.

InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

CouldNotQueryDatabase

The query required to get the database was not defined.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-AcctScopedObject -ObjectType Scheme
```

ScopelId : eff6f464-f1ee-4442-add3-99982e0cec01

ScopeName : Sales

ObjectType : Scheme

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-AcctService

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Removes metadata from the given IdentityPool.

Syntax

```
Remove-AcctIdentityPoolMetadata [-IdentityPoolUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]
```

```
Remove-AcctIdentityPoolMetadata [-IdentityPoolUid] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]
```

```
Remove-AcctIdentityPoolMetadata [-IdentityPoolName] <String> -Name <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]
```

```
Remove-AcctIdentityPoolMetadata [-IdentityPoolName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]
```

```
Remove-AcctIdentityPoolMetadata [-InputObject] <IdentityPool[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]
```

```
Remove-AcctIdentityPoolMetadata [-InputObject] <IdentityPool[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]
```

Detailed Description

Provides the ability to remove metadata from the given IdentityPool.

Related topics

[Set-AcctIdentityPoolMetadata](#)

Parameters

-IdentityPoolUid<Guid>

Id of the IdentityPool

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-IdentityPoolName<String>

Name of the IdentityPool

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<IdentityPool[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-

LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-AcctIdentityPool | % { Remove-AcctIdentityPoolMetadata -Map $_.MetadataMap }
```

Remove all metadata from all IdentityPool objects.

Remove-AcctIdentityPoolScope

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Resets the Active Directory machine password for the given accounts.

Syntax

```
Repair-AcctADAccount -ADAccountName <String[]> [-Password <String>] [-SecurePassword <SecureString>] [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Repair-AcctADAccount -ADAccountSid <String[]> [-Password <String>] [-SecurePassword <SecureString>] [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

This provides the ability to synchronize the account password stored in Active Directory with the password stored in the AD Identity Service. If successful, this results in the AD Identity Service account state being reset to 'available' so it can be consumed by other Machine Creation Services.

If the current account password is not supplied using the Password or SecurePassword Parameters, this requires the user who initiated the runspace to have the required permissions in Active Directory to reset the Active Directory Account password.

If the current account password is supplied then this command will use the password change operation which does not require any elevated permissions in Active Directory.

Related topics

[New-AcctADAccount](#)

[Add-AcctADAccount](#)

[Remove-AcctADAccount](#)

[Unlock-AcctADAccount](#)

[Update-AcctADAccount](#)

[Get-AcctADAccount](#)

Parameters

-ADAccount Name<String[]>

The Active Directory account name(s) that are to be repaired. Active Directory accounts are accepted in the following formats: Fully qualified DN e.g. CN=MyComputer,OU=Computers,DC=MyDomain,DC=Com; UPN format e.g. MyComputer@MyDomain.Com; Domain qualified e.g. MyDomain\MyComputer.

Required?	true
Default Value	

Accept Pipeline Input?	false

-ADAccountSid<String[]>

The Active Directory account SID(s) that are to be repaired.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Password<String>

The current password for the computer account.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SecurePassword<SecureString>

The current password for the account (provided in a Secure String class).

Required?	false
Default Value	
Accept Pipeline Input?	false

-Force<SwitchParameter>

Indicates whether accounts that are marked as 'in-use' can be repaired or not.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	LocalHost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

Citrix.AdIdentity.Sdk.AccountOperationSummary

The Repair-AcctADAccount command returns an object with the following parameters:

SuccessfulAccountsCount <int>

The number of accounts that were repaired successfully.

FailedAccountsCount <int>

The number of accounts that were not repaired.

FailedAccounts <Citrix.AdIdentity.Sdk.AccountError[]>

The list of accounts that failed to be repaired. Each one has the following parameters:

ADAccountName <string>

ADAccountSid <String>

ErrorReason <AdIdentityStatus>

This can be one of the following

UnableToConvertDomain

IdentityNotLocatedInDomain

IdentityNotFound

IdentityObjectInUse

IdentityObjectLocked

ADServiceDatabaseError

ADServiceDatabaseNotConfigured

ADServiceStatusInvalidDb

FailedToConnectToDomainController

FailedToExecuteSearchInAD

FailedToAccessComputerAccountInAD

FailedToSetPasswordInAD

FailedToChangePasswordInAD

DiagnosticInformation <Exception>

Any other error information

Notes

In the case of failure, the following errors can result.

Error Codes

PermissionDenied

The user does not have administrative rights to perform this operation.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with database failed for various reasons.

CommunicationError

An error occurred while communicating with the service.

ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

```
C:\PS>Repair-AcctADAccount -ADAccountName "Domain\account","domain\account2"
```

SuccessfulAccountsCount	FailedAccountsCount	FailedAccounts
2	0	{}

Reset-AcctServiceGroupMembership

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
Adds or updates metadata on the given IdentityPool.

Syntax

```
Set-AcctIdentityPoolMetadata [-IdentityPoolUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Set-AcctIdentityPoolMetadata [-IdentityPoolUid] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]

Set-AcctIdentityPoolMetadata [-IdentityPoolName] <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]

Set-AcctIdentityPoolMetadata [-IdentityPoolName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Set-AcctIdentityPoolMetadata [-InputObject] <IdentityPool[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Set-AcctIdentityPoolMetadata [-InputObject] <IdentityPool[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]
```

Detailed Description

Provides the ability for additional custom data to be stored against given IdentityPool objects.

Related topics

[Remove-AcctIdentityPoolMetadata](#)

Parameters

-IdentityPoolUid<Guid>

Id of the IdentityPool

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-IdentityPoolName<String>

Name of the IdentityPool

Required?	true

Default Value	
Accept Pipeline Input?	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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016

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

May 04, 2016
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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
Creates a new Citrix App-V policy containing the specified App-V Management and Publishing Server URLs.

Syntax

New-CtxAppVServer -PublishingServer <string> -ManagementServer <String> [-UserRefreshEnabled [<Boolean>]] [-UserRefreshOnLogon <Boolean>] [-UserRefreshInterval <int>] [-UserRefreshIntervalUnit <Enum>] [-GlobalRefreshEnabled [<Boolean>]] [-GlobalRefreshOnLogon <Boolean>] [-GlobalRefreshInterval <Int>] [-GlobalRefreshIntervalUnit <Enum>] [-CommonParameters< >]

Detailed Description

Creates a new Citrix App-V policy containing the specified App-V Management and Publishing server URLs. Additionally, accepts Publishing Server settings that control how and when automatic refresh occurs on the VDA.

Related topics

Parameters

-PublishingServer<>

URL of the Publishing server to add to the Citrix policy.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-ManagementServer<>

URL of the Management server to add to the Citrix policy.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-UserRefreshEnabled<>

Enables a refresh of packages published to user groups either at user logon or at a specified interval. For more information, see the App-V 5.0 documentation at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

Required?	false
Default Value	
Accept Pipeline Input?	false

-UserRefreshOnLogon<>

Specifies whether or not to initiate a refresh of packages published to user groups on every user logon. For more information, see the App-V 5.0 documentation at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

Required?	false
Default Value	
Accept Pipeline Input?	false

-UserRefreshInterval<>

Specifies the frequency at which to initiate a refresh of packages published to user groups. This can be either days or hours, as specified by the UserRefreshIntervalUnit setting. For more information, see the App-V 5.0 documentation at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

Required?	false
Default Value	
Accept Pipeline Input?	

Accept Pipeline Input?	false

-UserRefreshInterval<>

Specifies the frequency at which to initiate a refresh of packages published to user groups. This can be either days or hours, as specified by the UserRefreshIntervalUnit setting. For more information, see the App-V 5.0 documentation at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

Required?	false
Default Value	
Accept Pipeline Input?	false

-UserRefreshIntervalUnit<>

Specifies the unit for the UserRefreshInterval setting. This can be set to either Hours (0) or Days (1). For more information, see the App-V 5.0 documentation at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

Required?	false
Default Value	
Accept Pipeline Input?	false

-GlobalRefreshEnabled<>

Enables a refresh of packages published to machine groups either at user logon or at a specified interval. For more information, see the App-V 5.0 documentation at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

Required?	false
Default Value	
Accept Pipeline Input?	false

-GlobalRefreshOnLogon<>

Specifies whether or not to initiate a refresh of packages published to machine groups on every user logon. For more information, see the App-V 5.0 documentation at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

Required?	false
Default Value	
Accept Pipeline Input?	false

-GlobalRefreshInterval<>

Specifies the frequency at which to initiate a refresh of packages published to machine groups. This can be either days or hours, as specified by the GlobalRefreshIntervalUnit setting. Please refer to App-V 5.0 documentation for details. <http://technet.microsoft.com/en-us/library/jj687745.aspx>

Required?	false
Default Value	
Accept Pipeline Input?	false

-GlobalRefreshIntervalUnit<>

Specifies the unit for the GlobalRefreshInterval setting. This can be set to either Hours (0) or Days (1). Please refer to App-V 5.0 documentation for details. <http://technet.microsoft.com/en-us/library/jj687745.aspx>

Required?	false
Default Value	
Accept Pipeline Input?	false

Examples

EXAMPLE 1

New-CtxAppVServer -ManagementServer http://AppV-Mgmt-Server:8080 -PublishingServer http://appV-Pub-Server:8082

Creates a new Citrix Policy for Management Server AppV-Mgmt-Server:8080 & Publishing Server: AppV-Mgmt-Server:8082. Default Publishing Server setting is used for http://AppV-Mgmt-Server:8082

Default values for publishing server settings used for Http://AppV-Mgmt-Server:8082 are:

GlobalRefreshEnabled = false; GlobalLogonRefresh = false ; GlobalIntervalRefreshInterval = 0; GlobalIntervalRefreshUnit = Day

UserRefreshEnabled = true; UserLogonRefresh = true ; UserIntervalRefreshInterval = 0; GlobalIntervalRefreshUnit = Day

EXAMPLE 1

New-CtxAppVServer -ManagementServer http://AppV-Mgmt-Server:8080 -PublishingServer http://appV-Pub-Server:8082 -GlobalRefreshEnabled \$true -GlobalLogonRefresh \$true -GlobalRefreshInterval
Creates a new Citrix Policy for Management Server AppV-Mgmt-Server:8080 & Publishing Server: AppV-Mgmt-Server:8082. User specified Publishing Server settings are used for AppV-Mgmt-Server:8082

Following values are used to configure Publishing Server : http://AppV-Mgmt-Server:8082

GlobalRefreshEnabled = True; GlobalLogonRefresh = True ; GlobalIntervalRefreshInterval = 2; GlobalIntervalRefreshUnit = Hour

UserRefreshEnabled = true; UserLogonRefresh = true ; UserIntervalRefreshInterval = 3; GlobalIntervalRefreshUnit = Hour

Set-CtxAppVServerSetting

May 04, 2016
Specifies the App-V Publishing server settings to use on the VDA. These settings determine whether or not the App-V Client can automatically initiate a publishing refresh on certain events such as user logon or at specified intervals.

Syntax

Set-CtxAppVServerSetting -AppVPublishingServer <string> [-UserRefreshEnabled [<Boolean>]] [-UserRefreshOnLogon <Boolean>] [-UserRefreshInterval <int>] [-UserRefreshIntervalUnit <Enum>] [-GlobalRefreshEnabled [<Boolean>]] [-GlobalRefreshOnLogon <Boolean>] [-GlobalRefreshInterval <int>] [-GlobalRefreshIntervalUnit <Enum>] [<CommonParameters>]

Detailed Description

Specifies the App-V Publishing server settings to use on the VDA. These settings determine whether or not the App-V Client can automatically initiate a publishing refresh on certain events such as user logon or at specified intervals. Please refer to Microsoft App-V 5.0 documentation for more details on these settings : <http://technet.microsoft.com/en-us/library/jj687745.aspx>

Related topics

Parameters

-AppVPublishingServer<>

URL of the Publishing Server for which settings are to be set.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-UserRefreshEnabled<>

Enables a refresh of packages published to user groups either at user logon or at a specified interval. For more information, see the App-V documentation on the Microsoft website at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

Required?	True
Default Value	
Accept Pipeline Input?	false

-UserRefreshOnLogon<>

Specifies whether or not to initiate a refresh of packages published to user groups on every user logon. For more information, see the App-V documentation on the Microsoft website at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

Required?	true
Default Value	
Accept Pipeline Input?	false

-UserRefreshInterval<>

Specifies the frequency at which to initiate a refresh of packages published to user groups. This can be either days or hours, as specified by the UserRefreshIntervalUnit setting. For more information, see the App-V documentation on the Microsoft website at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

Required?	true
Default Value	
Accept Pipeline Input?	false

-UserRefreshIntervalUnit<>

Specifies the unit for the UserRefreshInterval setting. This can be set to either Hours (0) or Days (1). For more information, see the App-V documentation on the Microsoft website at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

Required?	true
Default Value	

Accept Pipeline Input?	false

-GlobalRefreshEnabled<>

Enables a refresh of packages published to machine groups either at user logon or at a specified interval. For more information, see the App-V documentation on the Microsoft website at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

Required?	True
Default Value	
Accept Pipeline Input?	false

-GlobalRefreshOnLogon<>

Specifies whether or not to initiate a refresh of packages published to machine groups on every user logon. For more information, see the App-V documentation on the Microsoft website at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

Required?	true
Default Value	
Accept Pipeline Input?	false

-GlobalRefreshInterval<>

Specifies the frequency at which to initiate a refresh of packages published to machine groups. This can be either days or hours, as specified by the GlobalRefreshIntervalUnit setting. For more information, see the App-V documentation on the Microsoft website at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

Required?	true
Default Value	
Accept Pipeline Input?	false

-GlobalRefreshIntervalUnit<>

Specifies the unit for the GlobalRefreshInterval setting. This can be set to either Hours (0) or Days (1). For more information, see the App-V documentation on the Microsoft website at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

Required?	true
Default Value	
Accept Pipeline Input?	false

Examples

----- EXAMPLE 1 -----

Set-CtxAppVServerSetting -AppVPublishingServer http://AppV50Publishing:8082 -GlobalRefreshEnabled \$true -GlobalRefreshOnLogon \$true -GlobalRefreshInterval 2 -GlobalRefreshIntervalUnit Hour -

Test-CtxAppVServer

May 04, 2016

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

May 04, 2016

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-BrokerAdminFolder	Get the admin folders in this site.
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.

Name	Description
Get-BrokerConfigurationSlot	Gets configuration slots configured for this site.
Get-BrokerConfiguredFTA	Gets any file type associations configured for an application.
Get-BrokerConnectionLog	Get entries from the site's session connection log.
Get-BrokerController	Gets Controllers running broker services in the site.
Get-BrokerDBConnection	Gets the database connection string for the specified data store used by the Broker Service.
Get-BrokerDBSchema	Gets SQL scripts to create or maintain the database schema for the Citrix Broker Service.
Get-BrokerDBVersionChangeScript	Gets an SQL service schema update script for the Citrix Broker Service.
Get-BrokerDelayedHostingPowerAction	Gets power actions that are executed after a delay.
Get-BrokerDesktop	Gets desktops configured for this site.
Get-BrokerDesktopGroup	Gets broker desktop groups configured for this site.
Get-BrokerDesktopUsage	Get usage history of desktop groups.
Get-BrokerEntitlementPolicyRule	Gets desktop rules from the site's entitlement policy.
Get-BrokerHostingPowerAction	Gets power actions queued for machines.
Get-BrokerHypervisorAlert	Gets hypervisor alerts recorded by the controller.
Get-BrokerHypervisorConnection	Gets hypervisor connections matching the specified criteria.
Get-BrokerIcon	Get stored icons.
Get-BrokerImportedFTA	Gets the imported file type associations.
Get-BrokerInstalledDbVersion	Gets a list of all available database schema versions for the Broker Service.
Get-BrokerLease	Gets stored leases.

Name	Description
Get-BrokerMachine	Gets machines belonging to this site.
Get-BrokerMachineCommand	Get the list of commands queued for delivery to a desktop.
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	Gets the current state of the Broker Service on the controller.
Get-BrokerSession	Gets a list of sessions.
Get-BrokerSessionLinger	Gets one or more session lingering settings.
Get-BrokerSessionPreLaunch	Gets one or more session pre-launch settings.
Get-BrokerSharedDesktop	Get shared desktops configured for this site.

Get-BrokerSite Name	Description
Get-BrokerTag	Gets one or more tags.
Get-BrokerUnconfiguredMachine	Gets machines that have registered but are not yet configured in this site.
Get-BrokerUser	Gets broker users configured for this site.
Group-BrokerDesktop	Groups and counts desktops with the same value for a specified property.
Group-BrokerMachine	Groups and counts machines with the same value for a specified property.
Group-BrokerSession	Groups and counts sessions with the same value for a specified property.
Import-BrokerDesktopPolicy	Sets the site wide Citrix Group Policy settings for the site.
Move-BrokerAdminFolder	Moves a folder to another place in the hierarchy, optionally renaming it
Move-BrokerApplication	Move a published application from one admin folder to another
New-BrokerAccessPolicyRule	Creates a new rule in the site's access policy.
New-BrokerAdminFolder	Creates a new admin folder.
New-BrokerAppAssignmentPolicyRule	Creates a new application rule in the site's assignment policy.
New-BrokerAppEntitlementPolicyRule	Creates a new application rule in the site's entitlement policy.
New-BrokerApplication	Creates a new published application.
New-BrokerAssignmentPolicyRule	Creates a new desktop rule in the site's assignment policy.
New-BrokerCatalog	Adds a new catalog to the site.
New-BrokerConfigurationSlot	Creates a new configuration slot.
New-BrokerConfiguredFTA	Creates a file type association with a published application.

Name	Description
New-BrokerDelayedHostingPowerAction	Causes a power action to be queued after a delay.
New-BrokerDesktopGroup	Create a new desktop group for managing the brokering of groups of desktops.
New-BrokerEntitlementPolicyRule	Creates a new desktop rule in the site's entitlement policy.
New-BrokerHostingPowerAction	Creates a new action in the power action queue.
New-BrokerHypervisorConnection	Creates a new hypervisor connection.
New-BrokerIcon	Creates a new icon.
New-BrokerMachine	Adds a machine that can be used to run desktops and applications.
New-BrokerMachineCommand	Creates a new command to deliver to a desktop.
New-BrokerMachineConfiguration	Creates a new machine configuration associated with an existing configuration slot.
New-BrokerPowerTimeScheme	Creates a new power time scheme for a desktop group.
New-BrokerRebootSchedule	Creates a new reboot schedule for a desktop group.
New-BrokerRemotePCAccount	Create a new RemotePCAccount.
New-BrokerSessionLinger	Creates a new session linger setting for a desktop group.
New-BrokerSessionPreLaunch	Creates a new session pre-launch setting for a desktop group.
New-BrokerTag	Creates a new tag.
New-BrokerUser	Creates a new broker user object
Remove-BrokerAccessPolicyRule	Deletes a rule from the site's access policy.
Remove-BrokerAccessPolicyRuleMetadata	Deletes AccessPolicyRule Metadata from the AccessPolicyRule objects
Remove-BrokerAdminFolder	Removes an admin folder.

Name	Description
Remove-BrokerAdminFolderMetadata	Deletes AdminFolder Metadata from the AdminFolder 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 objects
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

Name	Description
Remove-BrokerHostingPowerAction	EntitlementPolicyRule objects 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-BrokerLease	Remove the specified lease in the Database.
Remove-BrokerLeaseMetadata	Deletes Lease Metadata from the Lease objects
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
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 Name	Description
Remove-BrokerRebootSchedule	Removes the reboot schedule.
Remove-BrokerRemotePCAccount	Delete RemotePCAccounts from the system.
Remove-BrokerScope	Remove the specified catalog/desktop group from the given scope(s).
Remove-BrokerSessionLinger	Removes a session linger setting.
Remove-BrokerSessionMetadata	Deletes Session Metadata from the Session objects
Remove-BrokerSessionPreLaunch	Removes a session pre-launch setting.
Remove-BrokerSiteMetadata	Deletes Site Metadata from the Site objects
Remove-BrokerTag	Removes a tag from the system or clears a tag to object association.
Remove-BrokerTagMetadata	Deletes Tag Metadata from the Tag objects
Remove-BrokerUser	Remove broker user objects from another broker object
Rename-BrokerAccessPolicyRule	Renames a rule in the site's access policy.
Rename-BrokerAdminFolder	Renames a folder
Rename-BrokerAppAssignmentPolicyRule	Renames an application rule in the site's assignment policy.
Rename-BrokerAppEntitlementPolicyRule	Renames an application rule in the site's entitlement policy.
Rename-BrokerApplication	Renames an application.
Rename-BrokerAssignmentPolicyRule	Renames a desktop rule in the site's assignment policy.
Rename-BrokerCatalog	Renames a catalog.
Rename-BrokerDesktopGroup	Renames a desktop group.
Rename-BrokerEntitlementPolicyRule	Renames a desktop rule in the site's entitlement policy.

Name	Description
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	Reloads the access permissions and configuration service locations for the Broker Service.
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-BrokerAdminFolderMetadata	Creates/Updates metadata key-value pairs for AdminFolder
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.
Set-BrokerCatalogMetadata	Creates/Updates metadata key-value pairs for Catalog
Set-BrokerConfigurationSlotMetadata	Creates/Updates metadata key-value pairs for ConfigurationSlot

Name	Description
Set-BrokerControllerMetadata	Creates/Updates metadata key-value pairs for Controller
Set-BrokerDBConnection	Configures a database connection for the Broker Service.
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-BrokerLeaseMetadata	Creates/Updates metadata key-value pairs for Lease
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
Set-BrokerMachineMaintenanceMode	Sets whether the specified machine(s) are in maintenance mode.

Name	Description
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-BrokerSessionLinger	Updates the values of one or more desktop group session linger settings.
Set-BrokerSessionMetadata	Creates/Updates metadata key-value pairs for Session
Set-BrokerSessionPreLaunch	Updates the values of one or more desktop group session pre-launch settings.
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.

Name	Description
Start-BrokerRebootCycle	Creates and starts a reboot cycle for each desktop group that contains machines from the specified catalog.
Stop-BrokerRebootCycle	Cancels the specified reboot cycle.
Stop-BrokerSession	Stop or log off a session.
Test-BrokerAccessPolicyRuleNameAvailable	Determine whether the proposed AccessPolicyRule Name is available for use.
Test-BrokerAppAssignmentPolicyRuleNameAvailable	Determine whether the proposed AppAssignmentPolicyRule Name is available for use.
Test-BrokerAppEntitlementPolicyRuleNameAvailable	Determine whether the proposed AppEntitlementPolicyRule Name is available for use.
Test-BrokerApplicationNameAvailable	Determine whether the proposed Application Name is available for use.
Test-BrokerAssignmentPolicyRuleNameAvailable	Determine whether the proposed AssignmentPolicyRule Name is available for use.
Test-BrokerCatalogNameAvailable	Determine whether the proposed Catalog Name is available for use.
Test-BrokerDBConnection	Tests whether a database is suitable for use by the Citrix Broker Service.
Test-BrokerDesktopGroupNameAvailable	Determine whether the proposed DesktopGroup Name is available for use.
Test-BrokerEntitlementPolicyRuleNameAvailable	Determine whether the proposed EntitlementPolicyRule Name is available for use.
Test-BrokerLicenseServer	Tests whether or not a license server can be used by the broker.
Test-BrokerMachineNameAvailable	Determine whether the proposed Machine MachineName is available for use.
Test-BrokerPowerTimeSchemeNameAvailable	Determine whether the proposed PowerTimeScheme Name is available for use.
Test-BrokerRemotePCAccountNameAvailable	Determine whether the proposed RemotePCAccount OU is available for use.
Update-BrokerImportedFTA	Imports or updates all of the file type associations for the specified worker.
Update-BrokerLocalLeaseCache	Flushes the local lease cache.

Name	Description
e-BrokerNameCache	Administrative operations on the user and machine name cache.

about_Broker_AccessPolicy

May 04, 2016

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

May 04, 2016

TOPIC

Citrix Broker SDK - Applications

SHORT DESCRIPTION

Describes how to publish and manage hosted applications.

LONG DESCRIPTION

Published applications allow your users to launch applications as if they were installed on their devices when in fact they are hosted remotely. The applications are normally launched in a seamless window, meaning that users see only the application window and not an additional desktop.

Published applications are hosted on either desktop operating systems or server operating systems. Applications are published to users and desktop groups. As such, conceptually they exist "on top" of desktop groups, which are themselves built on top of catalogs. See about_Broker_Concepts for more information on catalogs.

There are two types of applications:

- o HostedOnDesktop - application runs on a remote machine and is displayed on the local client desktop.
- o InstalledOnClient - application is installed and run on the local client machine and has its window overlaid on to a remote desktop.

HOSTING APPLICATIONS

There are three main ways of hosting an application: using a private single-session VDI desktop, a shared single-session VDI desktop, and on shared multi-session server operating systems.

An application hosted on a shared single-session VDI desktop, when launched, is hosted on a random machine within the desktop group. An application hosted on a private single-session VDI desktop ensures that when a user launches the application it is always hosted on the same machine. An application hosted on shared multi-session server operating systems ensures that when a user launches the application it is always hosted on one of the least loaded servers in the desktop group.

You control how the application is hosted based on the kind of desktop group that you choose. Shared desktop groups randomly select a machine, and these desktop groups can only be created from catalogs with a Random AllocationType. On the other hand, private desktop groups host the application on the same machine for that user every time, and these desktop group types can only be created from catalogs with a Permanent AllocationType.

USER ACCESS & ASSIGNMENT

Users are not assigned an application directly, but instead they are required to first have access to the desktop groups on which the applications are published through access policy rules.

With shared desktop groups, access to a published application also needs an application entitlement policy rule. An application entitlement policy rule defines the set of users who are allowed per-session access to desktops in a specified desktop group.

With private desktop groups, access to a published application also needs an application assignment policy rule. An application assignment policy rule defines the set of users who are allowed access to a single application session in a desktop group.

Users are assigned private machines in one of two ways: pre-assigned or assign-on-first-use. Pre-assigned means that individual user accounts have been specified for the machines within that desktop group. A single machine can only have a single user account (not a group account) assigned to it, and likewise a user can only be assigned a single machine within a desktop group.

Assign-on-first-use requires less configuration. Machines are assigned to users the first time they log on. Rather than allocating users directly to machines, instead users are assigned to the private desktop group, either through individual user accounts or through user group accounts. Then, when a user assigned to the desktop group logs on, a machine is automatically allocated to them.

USER VISIBILITY

Users can be further filtered by configuring the visibility filter on top of the application. This restricts the application to a subset of the users that were granted access by the access policy and entitlement/assignment policy rules on the desktop group.

MULTIPLE DESKTOP GROUPS

You can publish an application to multiple desktop groups, which have to be of the same kind. Generally, an application that is published only to private desktop groups is referred to as a "private application". An application that is only published to shared desktop groups is referred to as a "shared application".

LICENSING

Hosted applications need the appropriate licenses to exist on the license server to be functional.

NAMING

Applications have three names that identify them: the Name, BrowserName and the PublishedName. The Name is unique for each application, is not visible to the user, and is primarily used for administrative purpose. The BrowserName is unique across the entire site, and is primarily used internally. The PublishedName is not unique and is the name seen by end users who have access to this application.

When creating an application, you only need to specify the Name. If no BrowserName is specified one is automatically generated. If no PublishedName is specified by default it is the same as the Name.

The following special characters are not allowed in the Name, BrowserName or the PublishedName properties: \ / ; # . * ? = < > | [] () " ' .

In addition the ` character is not allowed in the Name property.

To change the `PublishedName` or `BrowserName` of an application you must use the `Set-BrokerApplication` cmdlet.

To change the Name of an application you must use the `Rename-BrokerApplication` cmdlet.

OPTIONAL

You can configure file-type associations for applications, so that if a user double-clicks a document icon on their device, a published application automatically starts. For more information, see the help for `New-BrokerConfiguredFTA`.

You can apply tags to applications as another convenient way to further organize (and search for) applications. For more information, see the help for `New-BrokerTag`.

CMDLETS

`New-BrokerApplication`

Creates an application for publishing after the needed desktop groups, access policy and entitlement/assignment policy rules have been created.

`Add-BrokerApplication`

Adds one or more applications to a desktop group.

`Get-BrokerApplication`

Gets one or more applications.

`Remove-BrokerApplication`

Deletes one or more applications or it can be used to delete just the association of an application to a desktop group.

`Rename-BrokerApplication`

Changes the Name of an application.

`Set-BrokerApplication`

Changes the settings of application objects, except their names.

EXAMPLES

SHARED APPLICATION (SingleSession)

You have created a catalog with a Random AllocationType and SingleSession SessionSupport. It contains two machines called worker1 and worker2, both in the ACME domain. You publish an application with shared hosting as follows:

```
C:\PS> Write-Host "Create a desktop group, and add machines to it"
C:\PS> $dg = New-BrokerDesktopGroup "Shared Application Group"
-DesktopKind Shared -DeliveryType AppsOnly -SessionSupport 'SingleSession'
C:\PS> $m1 = Get-BrokerMachine -MachineName "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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

TOPIC

Citrix Broker SDK - Desktops

SHORT DESCRIPTION

Describes desktop concepts and usage.

LONG DESCRIPTION

A desktop is a machine that is able to run a Microsoft Windows desktop environment (with a shell, icons and taskbar) or individual applications (seamlessly integrated with the local desktop). The configuration of the desktop determines whether it can run only desktop environments, only applications, or both desktops and applications. Machines running workstation operating systems are able to run one session at a time (single-session), whereas machines running server operating systems have the ability to run multiple simultaneous sessions (multi-session).

A key aspect of desktops is how they are assigned (or allocated) to users. Two allocation types are supported:

- o Random/Shared - A user is assigned a desktop at random from a pool of shared desktops. Multi-session desktops are able to run sessions to multiple users simultaneously, whereas single-session desktops can only run one session at a time, and are returned to the pool when the user logs off. Single-session shared desktops usually discard user data stored on them after the user logs off. Multi-session shared desktops, however, do not tend to discard user data after a log-off, as this is only possible when the desktop is rebooted by a reboot schedule.
- o Permanent/Private - A private desktop is permanently assigned to a specific user and data stored on it is retained across logons and restarts. A private desktop can have users assigned explicitly or on first use.

DESKTOP GROUPS

Desktops are collected together in desktop groups, and these provide a flexible grouping mechanism that can be used to associate:

- o Desktops running on a particular type of machine
- o Desktops with particular software installed
- o Desktops for a set of users
- o Desktops accessed in a similar way
- o Desktops configured in a particular way
- o Any combination of the above

Each desktop group can only contain one type of desktop, determined by its `AllocationType` and `SessionSupport` properties.

When assigning shared desktops or assign-on-first-use (AOFU) desktops to users, the set of candidates comes from available desktops in a particular desktop group.

You configure power management policy for single-session desktop groups, including peak and off-peak settings, for each desktop group. See `about_Broker_PowerManagement` for details.

CREATION OF DESKTOPS

Desktop objects are created automatically when a machine is added to a desktop group. The type of desktop is determined by the `AllocationType` property of the desktop group.

In order for a machine to be added from a catalog, the machine must be compatible with the desktop group. For this to be true, the catalog's `AllocationType` must be compatible, and the `SessionSupport` property must match.

Note: Because the session support and functional level of the machine are determined by the software on the machine (operating system and Citrix VDA, respectively), the `SessionSupport` and `MinimalFunctionalLevel` of the catalog and desktop group may match, but not be compatible with the machine. In this case any attempt of registration by the machine will fail.

You can add machines explicitly using the `Add-BrokerMachine` cmdlet, or a number of free machines can be acquired from a catalog using the `Add-BrokerMachinesToDesktopGroup` cmdlet. A machine can only be associated with one desktop group at a time, and has a `DesktopUid` property that references the corresponding desktop object. You can also associate desktops to machines with their SID properties.

Desktop objects are deleted when the machine is removed from the desktop group (using the `Remove-BrokerMachine` cmdlet). Desktops are also deleted when the desktop group containing them is deleted (using the `Remove-BrokerDesktopGroup` cmdlet). A desktop cannot be removed from a catalog while it is in a desktop group.

SHARED (RANDOM) DESKTOPS

Shared desktops are published to users using entitlement policy rules. Each entitlement rule allows access to a single session on a desktop machine, selected at random from the available desktops in a desktop group (with a preference for desktops that are powered on). If there are no available desktops, launching the session fails. See `about_Broker_EntitlementPolicy` and `about_Broker_PowerManagement` for details.

PRIVATE (STATIC) DESKTOPS

You can assign private desktops to users explicitly or automatically, with the AOFU feature. It is also possible to assign private desktops to particular clients (through IP or client name).

You explicitly assign machines or desktops to users with the `Add-BrokerUser` cmdlet. Machines can be assigned to users before the machine has been added to the desktop group (desktop created), but otherwise the effect is the same.

With `Add-BrokerUser` you can assign a desktop to multiple users or user groups. If the desktop has single-session support then the desktop will be visible to multiple users, but only one user can log on to the desktop at any time.

Assignment policy rules allow you to use AOFU to assign desktops to users in a desktop group. When a user specified in an assignment policy rule launches a session, and if the user does not already have an assigned desktop, the broker selects an available desktop at random from the desktop group and permanently assigns it to that user. Once assigned in this way, the

desktop behaves as though the assignment was made explicitly by an administrator. See `about_Broker_AssignmentPolicy` for details.

It is possible to assign more than one desktop to a user. You can achieve this by explicit assignment, multiple assignment policy rules, or the `MaxDesktops` property of an assignment policy rule.

DESKTOP CONFIGURATION

When presented to the user, desktops are identified by:

- o An icon (`IconUid` property)
- o A name (`PublishedName` property)
- o A description (`Description` property)

When starting the session, you can configure two connection settings:

- o The color depth used at the start of the session (`ColorDepth` property)
- o Whether SecureICA encryption is required (`SecureIcaRequired` property)

Each desktop group provides default values for these settings, but you can override them if the desktop has more specific settings (`PrivateDesktop`), or use an entitlement policy rule with more specific settings (`SharedDesktop`).

AOFU desktops can inherit these settings from the assignment policy rule when the assignment to the user takes place.

MAINTENANCE MODE

There are times when it is necessary to disable desktops. You can do this by setting the `InMaintenanceMode` property of a desktop to `$true`. This puts it into maintenance mode. The broker excludes single-session desktops in maintenance mode from brokering decisions and does not start new sessions on them. Existing sessions are unaffected. For multi-session desktops in maintenance mode, reconnections to existing sessions are allowed, but no new sessions are created on the machine.

Desktops in maintenance mode are also excluded from automatic power management, although explicit power actions are still performed.

Note that disabling desktop groups, entitlement policy rules, assignment policy rules, or applications are other ways of disabling aspects of brokering.

DESKTOP STATUS

Once desktops are created, you can query the configuration and state information for different kinds of desktop, or retrieve more information about desktops using the `Get-BrokerMachine` cmdlet. To get details of any sessions running on the desktops, use the `Get-BrokerSession` cmdlet.

You can also group desktops by a specific property, counting the number of desktops with each value using the `Group-BrokerMachine` cmdlet. This can provide useful summary statistics.

DESKTOP USAGE

Every hour the broker records how many desktops from each desktop group are in use, and the `Get-BrokerDesktopUsage` cmdlet returns this information. Analyze historical usage records to understand desktop usage patterns and help with the choice of idle pool and buffer settings.

DESKTOP CONDITIONS

CPU usage, ICA latency, and profile logon times of desktops are monitored. When one of these values exceeds a threshold (configured by policy), the condition is flagged in the `DesktopConditions` property of the desktop. When the value drops below the threshold again, the condition is cleared. Use `Get-BrokerMachine` or `Group-BrokerMachines` cmdlets to query this information.

SEE ALSO

[about_Broker_Concepts](#)

[about_Broker_Applications](#)

[about_Broker_EntitlementPolicy](#)

[about_Broker_AssignmentPolicy](#)

[Add-BrokerMachine](#)

[Add-BrokerMachinesToDesktopGroup](#)

[Remove-BrokerMachine](#)

[Add-BrokerUser](#)

[Set-BrokerMachine](#)

[Set-BrokerPrivateDesktop](#)

[Get-BrokerMachine](#)

[Group-BrokerMachine](#)

[Get-BrokerDesktopUsage](#)

about_Broker_EntitlementPolicy

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

TOPIC

Citrix Broker SDK - Machine Power Management

SHORT DESCRIPTION

Describes power management of machines used for desktops and applications.

LONG DESCRIPTION

The Citrix Broker Service is in day-to-day control of the power state of the configured desktop and application machines. The Broker Service can control several hypervisors, each hypervisor connection being handled by its own site service, so all Broker Service communication to the hypervisor is through one of the controllers in the site.

HYPERVISOR CONNECTIONS

Each hypervisor, or pool of linked hypervisors, is described and configured through the XdHyp pseudo-drive and associated Hyp PowerShell commands (cmdlets) which are provided by the host service snap-in. When you have first created the hypervisor connection using the host service cmdlets, you can create a broker equivalent object that references the Hyp instance using a GUID value. Use the broker's HypervisorConnection object to nominate a preferred controller for direct communication with the hypervisor on behalf of all other controllers for day-to-day power actions and status requests.

POWER ACTIONS AND THROTTLING

The site, through the Broker Service, can control the power state of the machines used by the site for desktops and applications. Power state changes can have a number of causes:

- o Power policy rules, such as requests to shut down or suspend machines when user sessions on those machines end or are disconnected
- o If allowed, user-driven desktop restarts
- o Session launch requests requiring machines to be started
- o Pool size management, which controls the number of running machines
- o Direct administrator request using the SDK or Citrix Studio
- o Reboot schedules and cycles
- o Performing personal vDisk inventory activities
- o Cleaning machines back to the golden master image state after they have been used

The power state changes of machines hosting desktops and applications are controlled using a queuing mechanism. Actions to change the power state are assigned a priority and are sent to the hypervisor according to a throttling mechanism. This avoids overloading the hypervisor.

The queuing and throttling mechanisms take place on a per-hypervisor-connection basis; each hypervisor connection's queue is dealt with independently. You can view the contents of the queues using the Get-BrokerHostingPowerAction cmdlet. This includes recently completed, in-progress, and pending actions (that is, those due to be sent to the hypervisor

based on the throttling settings).

Each power action object comprises:

- o The machine to be acted on (Name, DNS Name, Hosting Name)
- o The action to be performed (TurnOn, TurnOff, Shutdown, Reset, Restart, Suspend, or Resume)
- o The action's priority (Base, the original priority, and Actual, the current priority)
- o The action's state (Pending, Started, Completed, Failed, Canceled, Deleted, or Lost)
- o Time stamps of the action's lifecycle points (when it was created, started, or completed)
- o Any reason the action failed

The throttling of power actions is controlled by three metadata values on the Hyp hypervisor connection object when accessed through the XdHyp pseudo-drive. The four values throttle power actions according to:

- o The maximum absolute number of in-progress power actions
- o The maximum number of in-progress power actions expressed as a percentage of the total number of machines controlled by the hypervisor connection
- o The maximum number of new power actions sent to the hypervisor per minute
- o The maximum number of in-progress PvD inventory activities expressed as a percentage of the total number of machines controlled by the hypervisor connection

You add power actions to the queue using the SDK's New-BrokerHostingPowerAction cmdlet. You cancel power actions in the queue using the Remove-BrokerHostingPowerAction cmdlet. You boost or reduce their priority using the Set-BrokerHostingPowerAction cmdlet.

You can also schedule power actions to be executed in the future, using the New-BrokerDelayedHostingPowerAction command. Only Shutdown and Suspend actions can be scheduled in this way. You can view these delayed power actions using Get-BrokerDelayedHostingPowerAction and cancel them with Remove-BrokerDelayedHostingPowerAction. When a delayed power action is ready to be executed it is deleted, and a corresponding normal power action is placed in the queue described above.

POWER POLICY

Policy rules associated with a desktop group allow you to change power states at configurable times after session state changes, typically a set number of minutes after session disconnection or session logout.

Note that these policy rules are defined directly as properties of the desktop group.

These policy actions allow the following operations to be specified:

- o A power action to be performed at a defined period after a session is disconnected
- o A power action to be performed at a defined extended period after a session is disconnected
- o A power action to be performed at a defined period after a session is logged off

The two disconnect policy actions are designed to allow multi-stage policies such as initially suspending a machine shortly after a session disconnect occurs, and then later powering-off the machine if the session has not been reconnected.

At the set time after the session state change, the required action is added to the power action queue, and this is then throttled and processed as normal.

POOL SIZE MANAGEMENT

You can manage flexibly the number of machines running desktops and applications using the pool size. For any given hour of the day and day of the week, this is an absolute number of machines or the percentage of the total number of machines in the desktop group. The pool size specifies the total number of machines that are always running, regardless of whether they are in use or idle. (Note: The number of machines does not depend on their idle status, but this does affect the buffer size value, which is also used to manage pool sizes.)

To start or shut down desktop machines to achieve the desired pool size, the system places power actions in the queue. Standard throttling queue management sends these to the hypervisors. A single desktop group (and its pool) can span multiple hypervisors, so actions to start and shut down machines can be added to multiple queues.

POWER TIME SCHEMES

Each single-session desktop group can be associated with one or more power time schemes, each scheme covering a number of days of the week. The time schemes specify, for each hour of the day, whether that hour is peak or off-peak. They also specify the number of running unassigned machines maintained by the broker.

You can configure other settings, such as buffer size and any power policy rules differently for peak and off-peak hours. You can define the number of running machines, idle or in use, either as an absolute value or as a percentage of the desktop group size. Machines running desktops and applications are started (or shut down when not in use) to match the required pool size.

Each power time scheme comprises:

- o The name of the scheme
- o The pattern of days of the week covered by the scheme
- o The set of hours considered peak and off peak
- o The set of pool size values (one for each hour of the day)

The hours of the day used by time schemes are the hours in the time zone for the desktop group the scheme is associated with. You cannot associate one desktop group with multiple time schemes covering the same day of the week.

BUFFER SIZE

In addition to the pool size, you can optionally configure two buffer sizes for each desktop group, one for peak hours and one for off-peak hours. The buffer size defines the minimum number of idle unassigned machines maintained by the broker and is specified as a percentage of the total machines in the group. These are running machines that are not used by any user session. The buffer size on its own never causes machines to shut down. It causes them to start up so a minimum number of idle machine is always available. The buffer size in conjunction with the pool size can cause machines running desktops or applications to be shut down.

POWER MANAGEMENT OF ASSIGNED MACHINES

Automatic power management for private desktop groups provides the ability to power on all assigned machines at the transition to a peak period and respectively power off all machines at the transition to an off-peak period.

If a machine is shut down during peak hours it will not be automatically powered on again, unless the `AutomaticPowerOnForAssignedDuringPeak` property on the desktop group is also enabled.

Note that all power management facilities apply only to single session machines.

REBOOT SCHEDULES

Reboot schedules are commonly used after image updates or to perform regular reboots of all machines in a desktop group or catalog to clear down problems resulting from a corrupt state or hung/faulty applications.

Reboot schedules allow distributing the reboot operation of all machines over a provided duration. Individual machine reboots are scheduled in a way that attempts to maintain maximum availability of machines in the group as the reboots occur, and avoid boot storms that overload the underlying infrastructure.

Reboot schedules are the only form of automatic power management that can shut down a machine while users are logged on; however the administrator can provide a warning message to be displayed to end users at a specified period prior to the shutdown taking effect.

REBOOT CYCLES

Reboot cycles describe the dynamic execution of desktop group or catalog reboot operations. Reboot cycles can be created due to reboot schedules, or by on-demand reboot operations requested via the SDK.

`RebootCycle` objects encapsulate the details of the associated reboot operation and can be queried to show the current status.

STATUS

You can view the status of the hypervisor connection on the broker hypervisor connection object. You can obtain any hypervisor alerts using the `Get-BrokerHypervisorAlert` cmdlet. You can check the power state of machines running desktops or applications using the relevant `Machine` or `Desktop` objects.

SEE ALSO

[about_Broker_Concepts](#)

[about_Broker_Machines](#)

[about_HypHostSnapin](#)

[New-BrokerHypervisorConnection](#)

Get-BrokerHypervisorConnection
Set-BrokerHypervisorConnection
Remove-BrokerHypervisorConnection
New-BrokerHostingPowerAction
Get-BrokerHostingPowerAction
Set-BrokerHostingPowerAction
Remove-BrokerHostingPowerAction
New-BrokerDelayedHostingPowerAction
Get-BrokerDelayedHostingPowerAction
Remove-BrokerDelayedHostingPowerAction
New-BrokerPowerTimeScheme
Get-BrokerPowerTimeScheme
Set-BrokerPowerTimeScheme
Rename-BrokerPowerTimeScheme
Remove-BrokerPowerTimeScheme
Get-BrokerRebootCycle
Set-BrokerRebootCycleMetadata
Start-BrokerRebootCycle
Stop-BrokerRebootCycle
Get-BrokerRebootSchedule
Set-BrokerRebootSchedule
New-BrokerRebootSchedule
Remove-BrokerRebootSchedule
New-BrokerDesktopGroup
Get-BrokerDesktopGroup
Set-BrokerDesktopGroup
Add-HypMetadata
Remove-HypMetadata

about_Broker_RemotePC

May 04, 2016

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

May 04, 2016
Adds applications to a desktop group.

Syntax

Add-BrokerApplication [-InputObject <Application[]> [-DesktopGroup <DesktopGroup>] [-Priority <Int32>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Add-BrokerApplication [-Name] <String> [-DesktopGroup <DesktopGroup>] [-Priority <Int32>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Detailed Description

The Add-BrokerApplication cmdlet is used to associate one or more applications with an existing desktop group.

There are two parameter sets for this cmdlet, allowing you to specify the application either by its BrowserName or by an array of object references. Uids can also be substituted for the object references.

See about_Broker_Desktops and about_Broker_Applications for more information.

Related topics

[New-BrokerApplication](#)

[Add-BrokerApplication](#)

[Add-BrokerTag](#)

[Remove-BrokerApplication](#)

[Rename-BrokerApplication](#)

[Move-BrokerApplication](#)

[Set-BrokerApplication](#)

Parameters

-InputObject<Application[]>

Specifies the application to associate. Its Uid can also be substituted for the object reference.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the application to be associated with the desktop group.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-DesktopGroup<DesktopGroup>

Specifies which desktop group this application should be associated with. Note that applications can only be associated with desktop groups of the AppsOnly or DesktopsAndApps delivery type.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByValue)

-Priority<Int32>

Specifies the priority of the mapping between the application and desktop group where lower numbers imply higher priority with zero being highest.

If one association has a higher priority than the other, machines from that group will be selected for launching sessions until all machines are at maximum load, in maintenance mode, unregistered, or unavailable for any other reason. Only when all machines from the higher-priority group are unavailable will new connections be routed to the next lowest priority group.

If multiple associations have equal priority, load balancing does not occur among the desktop groups in these associations. Instead, the broker chooses one of these groups as the preferred group and machines from this group will be selected for launching sessions until all machines are at maximum load, in maintenance mode, unregistered, or unavailable for any other reason. Only when all machines from the preferred group are unavailable will new connections be routed to another one of these groups, which the broker chooses as next-most preferred.

Required?	false
Default Value	0
Accept Pipeline Input?	true (ByPropertyName)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.Application, or as appropriate by property name You can pipe the application to be added to Add-BrokerApplication. You can also pipe some of the other parameters by name.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Add-BrokerApplication -BrowserName "Notepad" -DesktopGroup "Private DesktopGroup"
```

Adds the application with a BrowserName of "Notepad" to the desktop group called "Private DesktopGroup".

Add-BrokerDesktopGroup

May 04, 2016

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

May 04, 2016
Adds one or more machines to a desktop group.

Syntax

Add-BrokerMachine [-InputObject <Machine[]> [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Add-BrokerMachine [-MachineName <String> [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Detailed Description

The Add-BrokerMachine cmdlet adds specified machines to a desktop group. There are three forms:

- o Use the -InputObject parameter to add a single machine instance or array of instances to the group.
- o Use the -MachineName parameter to add a single, named machine to the group.
- o Use pipelining to pipe machines instances to the command.

The desktop group to which the machines are added can be specified by name, unique identifier (UID), or instance.

For a machine to be used in a site, the machine must be added to a desktop group. The machine and desktop group must be compatible in order for the process to succeed; for example a machine in a single-session catalog cannot be added to a multi-session desktop group.

For more information about machines, see about_Broker_Machines.

Related topics

[Add-BrokerMachinesToDesktopGroup](#)

[Remove-BrokerMachine](#)

[Get-BrokerMachine](#)

Parameters

-InputObject<Machine[]>

An array of machines to add to the group.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-MachineName<String>

The name of the single machine to add (must match the MachineName property of the machine).

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-DesktopGroup<DesktopGroup>

The desktop group to which the machines are added, specified by name, Uid, or instance.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.Machine You can pipe in the machines you want to add.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Add-BrokerMachine -InputObject $machine -DesktopGroup $desktopGroup
```

```
C:\PS> Add-BrokerMachine -InputObject $machine -DesktopGroup 2
```

```
C:\PS> Add-BrokerMachine $machine -DesktopGroup "MyDesktopGroup"
```

These examples all add a single machine instance to a desktop group, identifying the group by instance, UID, or name.

----- EXAMPLE 2 -----

```
C:\PS> Add-BrokerMachine -MachineName "MyDomain\MyMachine" -DesktopGroup 2
```

```
C:\PS> Add-BrokerMachine "MyDomain\MyMachine" -DesktopGroup "MyDesktopGroup"
```

```
C:\PS> Add-BrokerMachine "MyDomain\MyMachine" -DesktopGroup $desktopGroup
```

These examples add the machine called MyMachine to a desktop group.

----- EXAMPLE 3 -----

```
C:\PS> Get-BrokerMachine -Uid 3 | Add-BrokerMachine -DesktopGroup 2
```

```
C:\PS> Get-BrokerMachine -CatalogUid 4 | Add-BrokerMachine -DesktopGroup 2
```

These examples find specific machines and add them to a desktop group.

Add-BrokerMachineConfiguration

May 04, 2016
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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
在用户与另一个代理对象之间创建关联

语法

```
Add-BrokerUser [-InputObject] [-Application] [-SessionLinger] [-SessionPreLaunch] [-Machine] [-PrivateDesktop] [-LoggingId] [-AdminAddress] []  
  
Add-BrokerUser [-Name] [-Application] [-SessionLinger] [-SessionPreLaunch] [-Machine] [-PrivateDesktop] [-LoggingId] [-AdminAddress] []
```

详细说明

Add-BrokerUser cmdlet 向另一个指定对象（如代理专用桌面）添加代理用户对象。这取决于模板对象类型：

- o 计算机 - 向指定用户分配代理计算机；之后将计算机添加到桌面组时，桌面还分配给相同用户。
- o 专用桌面 - 向指定用户分配桌面。
- o 应用程序 - 向指定用户分配应用程序。

相关主题

[Get-BrokerUser](#)
[Remove-BrokerUser](#)

参数

-InputObject

要添加的用户对象。

是否必需？	true
默认值	空值
是否接受管道输入？	true (ByValue)

-Name

要添加的一个或多个用户的名称。

是否必需？	true
默认值	空值

是否接受管道输入?	true (ByPropertyName)

-Application

要与用户关联的应用程序。

是否必需?	false
默认值	
是否接受管道输入?	true (ByValue)

-SessionLinger

要分配给用户的会话延迟设置。

是否必需?	false
默认值	空值
是否接受管道输入?	true (ByValue)

-SessionPreLaunch

要分配给用户的会话预启动设置。

是否必需?	false
默认值	空值
是否接受管道输入?	true (ByValue)

-Machine

要分配给用户的计算机

是否必需?	false
默认值	空值
是否接受管道输入?	true (ByValue)

-PrivateDesktop

要分配给用户的桌面

是否必需？	false
默认值	空值
是否接受管道输入？	true (ByValue)

-LoggingId

指定此 cmdlet 调用所属的高级操作的标识符。Desktop Studio 和 Desktop Director 通常创建高级操作。PowerShell 脚本也可以借助 Start-LogHighLevelOperation 和 Stop-LogHighLevelOperation cmdlet 在高级操作中打包一系列 cmdlet 调用。

是否必需？	false
默认值	
是否接受管道输入？	false

-AdminAddress

指定 PowerShell 管理单元将要连接的 XenDesktop 控制器的地址。可以提供主机名或 IP 地址。

是否必需？	false
默认值	Localhost。一旦有 cmdlet 提供了某个值，此值将变为默认值。
是否接受管道输入？	false

输入类型

Citrix.Broker.Admin.SDK.User 您可以将要添加的用户通过管道传递给 Add-BrokerUser。

返回值

无

备注

仅指定 -Machine、-PrivateDesktop 或 -Application 参数中的一个。

示例

----- 示例 1 -----

```
Add-BrokerUser "DOMAIN\UserName" -PrivateDesktop "DOMAIN\MachineName"
```

向指定用户分配指定专用桌面。

----- 示例 2 -----

```
Add-BrokerUser "DOMAIN\UserName" -Application "ApplicationName"
```

向指定用户分配指定应用程序。

----- 示例 3 -----

```
Add-BrokerUser "DOMAIN\UserName" -Application "FolderName"\ApplicationName"
```

向指定用户分配指定文件夹中的指定应用程序。

Disconnect-BrokerSession

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Gets rules from the site's access policy.

Syntax

```
Get-BrokerAccessPolicyRule [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerAccessPolicyRule [-Name] <String> [-AllowedConnections <AllowedConnection>] [-AllowedUsers  
<AllowedUser>] [-Description <String>] [-DesktopGroupName <String>] [-DesktopGroupUid <Int32>] [-Enabled <Boolean>] [-  
ExcludedClientIPFilterEnabled <Boolean>] [-ExcludedClientName <String>] [-ExcludedClientNameFilterEnabled <Boolean>] [-  
ExcludedSmartAccessFilterEnabled <Boolean>] [-ExcludedSmartAccessTag <String>] [-ExcludedUser <User>] [-  
ExcludedUserFilterEnabled <Boolean>] [-IncludedClientIPFilterEnabled <Boolean>] [-IncludedClientName <String>] [-  
IncludedClientNameFilterEnabled <Boolean>] [-IncludedSmartAccessFilterEnabled <Boolean>] [-IncludedSmartAccessTag  
<String>] [-IncludedUser <User>] [-IncludedUserFilterEnabled <Boolean>] [-Metadata <String>] [-ReturnTotalRecordCount] [-  
MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

Returns rules matching the specified search criteria from the site's access policy. If no search criteria are specified, all rules in the access policy are obtained.

An access policy rule defines a set of connection filters and access control rights relating to a desktop group. These allow fine-grained control of what access is granted to a desktop group based on details of, for example, a user's endpoint device, its address, and the user's identity.

----- BrokerAccessPolicyRule Object

A BrokerAccessPolicyRule object represents a single rule within the site's access policy. For a user to gain access to a desktop group via the rule their connection must match all its enabled include filters, and none of its enabled exclude filters. The object contains the following properties:

-- AllowedConnections (Citrix.Broker.Admin.SDK.AllowedConnection)

Controls whether connections must be local or via Access Gateway, and if so whether specified SmartAccess tags must be provided by Access Gateway with the connection. This property forms part of the included SmartAccess tags filter.

For a detailed description of this property see "help about_Broker_AccessPolicy".

-- AllowedProtocols (System.String[])

Protocols (for example HDX, RDP) available to the user for sessions delivered from the rule's desktop group. If the user gains access to a desktop group by multiple rules, the allowed protocol list is the combination of the protocol lists from all those rules.

If the protocol list is empty, access to the desktop group is implicitly denied.

-- AllowedUsers (Citrix.Broker.Admin.SDK.AllowedUser)

Controls the behavior of the included users filter. This can restrict access to a list of named users or groups, or allow access to any authenticated user. For a detailed description of this property see "help about_Broker_AccessPolicy".

-- AllowRestart (System.Boolean)

Indicates if the user can restart sessions delivered from the rule's desktop group. Session restart is handled as follows: For sessions on single-session power-managed machines, the machine is powered off, and a new session launch request made; for sessions on multi-session machines, a logoff request is issued to the session, and a new session launch request made; otherwise the property is ignored.

-- Description (System.String)

An optional description of the rule. The text is purely informational for the administrator, it is never visible to the end user.

-- DesktopGroupName (System.String)

The name of the desktop group to which the rule applies.

-- DesktopGroupUid (System.Int32)

The unique ID of the desktop group to which the rule applies.

-- Enabled (System.Boolean)

Indicates whether the rule is enabled. A disabled rule is ignored when evaluating the site's access policy.

-- ExcludedClientIPFilterEnabled (System.Boolean)

Indicates whether the excluded client IP filter is enabled. If the filter is disabled it is ignored when the rule is evaluated.

-- ExcludedClientIPs (Citrix.Broker.Admin.SDK.ChbIPAddressRange[])

IP addresses of user devices explicitly denied access to the rule's desktop group. Addresses can be specified as simple numeric addresses or as subnet masks (for example, 10.40.37.5 or 10.40.0.0/16). This property forms part of the excluded client IP address filter.

-- ExcludedClientNameFilterEnabled (System.Boolean)

Indicates whether the excluded client name filter is enabled. If the filter is disabled it is ignored when the rule is evaluated.

-- ExcludedClientNames (System.String[])

Names of user devices explicitly denied access to the rule's desktop group. This property forms part of the excluded client names filter.

-- ExcludedSmartAccessFilterEnabled (System.Boolean)

Indicates whether the excluded SmartAccess tags filter is enabled. If the filter is disabled it is ignored when the rule is evaluated.

-- ExcludedSmartAccessTags (System.String[])

SmartAccess tags which explicitly deny access to the rule's desktop group if any occur in those provided by Access Gateway with the user's connection. This property forms part of the excluded SmartAccess tags filter.

-- ExcludedUserFilterEnabled (System.Boolean)

Indicates whether the excluded users filter is enabled. If the filter is disabled it is ignored when the rule is evaluated.

-- ExcludedUsers (Citrix.Broker.Admin.SDK.ChbUser[])

Users and groups who are explicitly denied access to the rule's desktop group. This property forms part of the excluded users filter.

-- HdxSslEnabled (System.Boolean)

Indicates whether SSL encryption is enabled for sessions delivered from the rule's desktop group.

-- IncludedClientIPFilterEnabled (System.Boolean)

Indicates whether the included client IP filter is enabled. If the filter is disabled it is ignored when the rule is evaluated.

-- IncludedClientIPs (Citrix.Broker.Admin.SDK.ChbIPAddressRange[])

IP addresses of user devices allowed access to the rule's desktop group. Addresses can be specified as simple numeric addresses or as subnet masks (for example, 10.40.37.5 or 10.40.0.0/16). This property forms part of the included client IP address filter.

-- IncludedClientNameFilterEnabled (System.Boolean)

Indicates whether the included client names filter is enabled. If the filter is disabled it is ignored when the rule is evaluated.

-- IncludedClientNames (System.String[])

Names of user devices allowed access to the rule's desktop group. This property forms part of the included client names filter.

-- IncludedSmartAccessFilterEnabled (System.Boolean)

Indicates whether the included SmartAccess tags filter is enabled. If the filter is disabled it is ignored when the rule is evaluated.

-- IncludedSmartAccessTags (System.String[])

The SmartAccess tags which grant access to the rule's desktop group if any occur in those provided by Access Gateway with the user's connection. This property forms part of the excluded SmartAccess tags filter.

-- IncludedUserFilterEnabled (System.Boolean)

Indicates whether the included users filter is enabled. If the filter is disabled it is ignored when the rule is evaluated.

-- IncludedUsers (Citrix.Broker.Admin.SDK.ChbUser[])

Users and groups who are granted access to the rule's desktop group. This property forms part of the included users filter.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

A collection of arbitrary key/value pairs that can be associated with the rule. The administrator can use these values for any purpose; they are not used by the site itself in any way.

-- Name (System.String)

Administrative name of the rule. Each rule in the site's access policy must have a unique name.

-- Uid (System.Int32)

Unique ID of the rule itself.

Related topics

[New-BrokerAccessPolicyRule](#)

[Set-BrokerAccessPolicyRule](#)

[Rename-BrokerAccessPolicyRule](#)

[Remove-BrokerAccessPolicyRule](#)

Parameters

-Uid<Int32>

Gets only the rule with the specified unique ID.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Name<String>

Gets only rules with the specified name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AllowedConnections<AllowedConnection>

Gets only rules that have the specified value in the AllowedConnections property of their included SmartAccess tags filter.

Valid values are Filtered, NotViaAG, and ViaAG.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AllowedUsers<AllowedUser>

Gets only rules that have the specified value in the AllowedUsers property of their included users filter.

Valid values are Filtered, AnyAuthenticated, Any, AnonymousOnly and FilteredOrAnonymous.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Description<String>

Gets only rules with the specified description.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopGroupName<String>

Gets only rules applying to desktop groups with names matching the specified name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopGroupUid<Int32>

Gets only rules that apply to the desktop group with the specified unique ID.

Required?	false

Default Value	
Accept Pipeline Input?	false

-Enabled<Boolean>

Gets only rules that are in the specified state, either enabled (\$true) or disabled (\$false).

Required?	false
Default Value	
Accept Pipeline Input?	false

-ExcludedClientIPFilterEnabled<Boolean>

Gets only rules that have their excluded client IP address filter enabled (\$true) or disabled (\$false).

Required?	false
Default Value	
Accept Pipeline Input?	false

-ExcludedClientName<String>

Gets only rules that have the specified client name in their excluded client names filter (whether the filter is enabled or not).

Required?	false
Default Value	
Accept Pipeline Input?	false

-ExcludedClientNameFilterEnabled<Boolean>

Gets only rules that have their excluded client name filter enabled (\$true) or disabled (\$false).

Required?	false
Default Value	

Accept Pipeline Input?	false

-ExcludedSmartAccessFilterEnabled<Boolean>

Gets only rules that have their excluded SmartAccess tags filter enabled (\$true) or disabled (\$false).

Required?	false
Default Value	
Accept Pipeline Input?	false

-ExcludedSmartAccessTag<String>

Gets only rules that have the specified SmartAccess tag in their excluded SmartAccess tags filter (whether the filter is enabled or not).

Required?	false
Default Value	
Accept Pipeline Input?	false

-ExcludedUser<User>

Gets only rules that have the specified user in their excluded users filter (whether the filter is enabled or not).

Required?	false
Default Value	
Accept Pipeline Input?	false

-ExcludedUserFilterEnabled<Boolean>

Gets only rules that have their excluded user filter enabled (\$true) or disabled (\$false).

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedClientIPFilterEnabled<Boolean>

Gets only rules that have their included client IP address filter enabled (\$true) or disabled (\$false).

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedClientName<String>

Gets only rules that have the specified user device name in their included client names filter (whether the filter is enabled or not).

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedClientNameFilterEnabled<Boolean>

Gets only rules that have their included client name filter enabled (\$true) or disabled (\$false).

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedSmartAccessFilterEnabled<Boolean>

Gets only rules that have their included SmartAccess tags filter enabled (\$true) or disabled (\$false).

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedSmartAccessTag<String>

Gets only rules that have the specified SmartAccess tag in their included SmartAccess tags filter (whether the filter is enabled or not).

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedUser<User>

Gets only rules that have the specified user in their included users filter (whether the filter is enabled or not).

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedUserFilterEnabled<Boolean>

Gets only rules that have their included user filter enabled (\$true) or disabled (\$false).

Required?	false
Default Value	
Accept Pipeline Input?	false

-Metadata<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

Required?	false
Default Value	

Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about_Broker_Filtering for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false

Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell style filter expression. See about_Broker_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.AccessPolicyRule

Get-BrokerAccessPolicyRule returns all access policy rules that match the specified selection criteria.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Get-BrokerAccessPolicyRule
```

Returns all access policy rules. This offers a complete description of the current site's access policy.

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerAccessPolicyRule -Enabled $true -IncludedUser sales\tech-support
```

Returns all rules that are both enabled and explicitly include the SALES\tech-support group in their included users filter.

Get-BrokerAdminFolder

May 04, 2016

Get the admin folders in this site.

Syntax

```
Get-BrokerAdminFolder [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerAdminFolder [[-Name] <String>] [-DirectChildAdminFolders <Int32>] [-DirectChildApplications <Int32>] [-FolderName <String>] [-LastChangeld <Guid>] [-Metadata <String>] [-ParentAdminFolderUid <Int32>] [-TotalChildApplications <Int32>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Get-BrokerAdminFolder cmdlet gets admin folders in this site.

Without parameters, Get-BrokerAdminFolder gets all the admin folders that have been created. You can also use the parameters of Get-BrokerAdminFolder to filter the results to just the folders you're interested in. You can also identify folders by their UIDs or their FolderNames.

----- BrokerAdminFolder Object

A folder for use in the administration console for organising other objects. E.g. BrokerApplication objects

-- DirectChildAdminFolders (System.Int32)

The number of admin folders with this folder as a direct parent

-- DirectChildApplications (System.Int32)

The number of applications in this admin folder (does not include any applications in child folders)

-- FolderName (System.String)

The simple name of this folder within any parent folder

-- LastChangeld (System.Guid)

A random GUID assigned whenever there is a change anywhere in the hierarchy of admin folders below this node; each change updates this value on the changed folder and all parents all the way up to the root folder. Note that nodes below any change do not have their LastChangeld value updated

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Holds any metadata associated with the admin folder

-- Name (System.String)

The full name of the folder including the full parent hierarchy separated by back-slash characters and including a trailing back-slash

-- ParentAdminFolderUid (System.Int32)

The UID of the parent admin folder; the root folder references itself (zero)

-- TotalChildApplications (System.Int32)

The number of applications in this admin folder (including any applications in child folders)

-- Uid (System.Int32)

The unique ID of the admin folder (the root folder has the value zero)

Related topics

[New-BrokerAdminFolder](#)

[Remove-BrokerAdminFolder](#)

Parameters

-Uid<Int32>

Gets only the admin folder with the specified unique identifier.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Name<String>

Gets admin folders matching the specified name (if no trailing backslash is supplied, it is assumed).

Required?	false
Default Value	
Accept Pipeline Input?	false

-DirectChildAdminFolders<Int32>

Gets admin folders with the specified number of child folders.

Required?	false
Default Value	

Accept Pipeline Input?	false

-DirectChildApplications<Int32>

Gets admin folders with the specified number of applications (excluding those in sub-folders).

Required?	false
Default Value	
Accept Pipeline Input?	false

-FolderName<String>

Gets only the admin folders matching the specified simple folder name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastChangeId<Guid>

Gets only the admin folders with the specified value for LastChangeId.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Metadata<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

Required?	false
Default Value	

Accept Pipeline Input?	false

-ParentAdminFolderUid<Int32>

Gets only admin folders with the specified parent admin folder UID value.

Required?	false
Default Value	
Accept Pipeline Input?	false

-TotalChildApplications<Int32>

Gets admin folders with the specified number of applications (including those in sub-folders).

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about_Broker_Filtering](#) for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250

Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell style filter expression. See about_Broker_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None Input cannot be piped to this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.AdminFolder

Returns admin folders.

Examples

----- **EXAMPLE 1** -----

C:\PS> Get-BrokerAdminFolder
Return all administration folders.

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerAdminFolder -Uid 1
Get the administration folder with Uid 1.

Get-BrokerAppAssignmentPolicyRule

May 04, 2016

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

May 04, 2016

Gets application rules from the site's entitlement policy.

Syntax

```
Get-BrokerAppEntitlementPolicyRule [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerAppEntitlementPolicyRule [[-Name] <String>] [-Description <String>] [-DesktopGroupUid <Int32>] [-Enabled  
<Boolean>] [-ExcludedUser <User>] [-ExcludedUserFilterEnabled <Boolean>] [-IncludedUser <User>] [-  
IncludedUserFilterEnabled <Boolean>] [-SessionReconnection <SessionReconnection>] [-ReturnTotalRecordCount] [-  
MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

Returns application rules matching the specified search criteria from the site's entitlement policy. If no search criteria are specified, all application rules in the entitlement policy are obtained.

An application rule in the entitlement policy defines the users who are allowed per-session access to a machine to run one or more applications published from the rule's desktop group.

----- BrokerAppEntitlementPolicyRule Object

The BrokerAppEntitlementPolicyRule object represents a single application rule within the site's entitlement policy. It contains the following properties:

-- Description (System.String)

Optional description of the rule. The text is purely informational for the administrator, it is never visible to the end user.

-- DesktopGroupUid (System.Int32)

The unique ID of the desktop group to which the rule applies.

-- Enabled (System.Boolean)

Indicates whether the rule is enabled. A disabled rule is ignored when evaluating the site's entitlement policy.

-- ExcludedUserFilterEnabled (System.Boolean)

Indicates whether the excluded users filter of the rule is enabled. If the filter is disabled then any user entries in the filter are ignored when entitlement policy rules are evaluated.

-- ExcludedUsers (Citrix.Broker.Admin.SDK.ChbUser[])

The excluded users filter of the rule, that is, the users and groups who are explicitly denied entitlements to use published applications from the associated desktop group.

-- IncludedUserFilterEnabled (System.Boolean)

Indicates whether the included users filter of the rule is enabled. If the filter is disabled then any user who satisfies the

requirements of the access policy is implicitly granted an entitlement to an application session by the rule.

-- IncludedUsers (Citrix.Broker.Admin.SDK.ChbUser[])

The included users filter of the rule, that is, the users and groups who are granted an entitlement to an application session by the rule.

If a user appears explicitly in the excluded users filter of the rule or is a member of a group that appears in the excluded users filter, no entitlement is granted whether or not the user appears in the included users filter.

-- Name (System.String)

The administrative name of the rule. Each rule in the site's entitlement policy must have a unique name (irrespective of whether they are desktop or application rules).

-- SessionReconnection (Citrix.Broker.Admin.SDK.SessionReconnection)

Defines reconnection (roaming) behavior for sessions launched using this rule. Session reconnection control is an experimental and unsupported feature.

-- Uid (System.Int32)

The unique ID of the rule itself.

Related topics

[New-BrokerAppEntitlementPolicyRule](#)

[Set-BrokerAppEntitlementPolicyRule](#)

[Rename-BrokerAppEntitlementPolicyRule](#)

[Remove-BrokerAppEntitlementPolicyRule](#)

Parameters

-Uid<Int32>

Gets the application rule with the specified unique ID.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Name<String>

Gets only application rules with the specified name.

Required?	false

Default Value	
Accept Pipeline Input?	false

-Description<String>

Gets only application rules with the specified description.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopGroupUid<Int32>

Gets only the application rule that applies to the desktop group with the specified unique ID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Enabled<Boolean>

Gets only application rules that are in the specified state, either enabled (\$true), or disabled (\$false).

Required?	false
Default Value	
Accept Pipeline Input?	false

-ExcludedUser<User>

Gets only application rules that have the specified user in their excluded users filter (whether the filter is enabled or not).

Required?	false
Default Value	

Accept Pipeline Input?	false

-ExcludedUserFilterEnabled<Boolean>

Gets only application rules that have their excluded user filter enabled (\$true) or disabled (\$false).

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedUser<User>

Gets only application rules that have the specified user in their included users filter (whether the filter is enabled or not).

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedUserFilterEnabled<Boolean>

Gets only application rules that have their included user filter enabled (\$true) or disabled (\$false).

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionReconnection<SessionReconnection>

Gets only application rules with the specified session reconnection behavior.

Required?	false
Default Value	

Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about_Broker_Filtering for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false

Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell style filter expression. See [about_Broker_Filtering](#) for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through `Select-Object`, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.AppEntitlementPolicyRule

Get-BrokerAppEntitlementPolicyRule returns all application rules from the entitlement policy that match the specified selection criteria.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Get-BrokerAppEntitlementPolicyRule
```

Returns all application rules from the entitlement policy. This offers a complete description of the current site's entitlement policy with respect to application entitlements from shared desktop groups.

----- EXAMPLE 2 -----

```
C:\PS> $dg = Get-BrokerDesktopGroup 'Customer Support'
```

```
C:\PS> Get-BrokerAppEntitlementPolicyRule -DesktopGroupUid $dg.Uid
```

Returns the application rule in the entitlement policy that grants users an entitlement to application sessions in the Customer Support desktop group.

Get-BrokerApplication

May 04, 2016

Get the applications published on this site.

Syntax

```
Get-BrokerApplication [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerApplication [-Name] <String> [-AdminFolderName <String>] [-AdminFolderUid <Int32>] [-ApplicationName  
<String>] [-ApplicationType <ApplicationType>] [-AssociatedDesktopGroupPriority <Int32>] [-AssociatedDesktopGroupUid  
<Int32>] [-AssociatedDesktopGroupUUID <Guid>] [-AssociatedUserFullName <String>] [-AssociatedUserName <String>] [-  
AssociatedUserUPN <String>] [-BrowserName <String>] [-ClientFolder <String>] [-CommandLineArguments <String>] [-  
CommandLineExecutable <String>] [-CpuPriorityLevel <CpuPriorityLevel>] [-Description <String>] [-Enabled <Boolean>] [-  
IconFromClient <Boolean>] [-IconUid <Int32>] [-MetadataKey <String>] [-Metadata <String>] [-PublishedName <String>] [-  
SecureCmdLineArgumentsEnabled <Boolean>] [-ShortcutAddedToDesktop <Boolean>] [-ShortcutAddedToStartMenu  
<Boolean>] [-StartMenuFolder <String>] [-UserFilterEnabled <Boolean>] [-UUID <Guid>] [-Visible <Boolean>] [-  
WaitForPrinterCreation <Boolean>] [-WorkingDirectory <String>] [-DesktopUid <Int32>] [-SessionUid <Int64>] [-UserSID  
<String>] [-DesktopGroupUid <Int32>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy  
<String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Get-BrokerApplication cmdlet gets the published applications that are hosted on this site.

Without parameters, Get-BrokerApplication gets all the applications that have been published, regardless of whether they are visible to users or not. You can also use the parameters of Get-BrokerApplication to filter the results to just the applications you're interested in. You can also identify applications by their UIDs or their BrowserNames.

For more information about applications, see about_Broker_Applications.

----- BrokerApplication Object

The BrokerApplication object represents a published application in the site. It contains the following properties:

-- AdminFolderName (System.String)

The name of the admin folder the application is in (including trailing backslash), or the empty string if the application is at the root level

-- AdminFolderUid (System.Int32)

The Uid of the admin folder the application is in (if any)

-- ApplicationName (System.String)

The simple name of the application within its parent admin folder (if any)

-- ApplicationType (Citrix.Broker.Admin.SDK.ApplicationType)

The type of the application, whether HostedOnDesktop or InstalledOnClient.

-- AssociatedDesktopGroupPriorities (System.Int32[])

List of associated desktop group priorities. Associated desktop groups is the list of desktop groups on which the application is published. When launching an application an available machine from one of the associated groups is selected. Desktop groups are searched for available machines in order of their priority.

-- AssociatedDesktopGroupUids (System.Int32[])

List of associated desktop group uids. Associated desktop groups is the list of desktop groups on which the application is published. The list is sorted by priority, with the highest priority group first.

-- AssociatedDesktopGroupUUIDs (System.Guid[])

List of associated desktop group UUIDs. Associated desktop groups is the list of desktop groups on which the application is published. The list is sorted by priority, with the highest priority group first.

-- AssociatedUserFullNames (System.String[])

List of associated users (full names). Associated users is the list of users who are given access using the application/user mapping filter.

-- AssociatedUserNames (System.String[])

List of associated users (SAM names). Associated users is the list of users who are given access using the application/user mapping filter.

-- AssociatedUserUPNs (System.String[])

List of associated users (user principle names). Associated users is the list of users who are given access using the application/user mapping filter.

-- BrowserName (System.String)

Unique browser name used to identify this application to other components in the site. This value is not visible to the end users.

-- ClientFolder (System.String)

The folder that the application belongs to as the user sees it.

-- CommandLineArguments (System.String)

The command-line arguments to use when launching the executable.

-- CommandLineExecutable (System.String)

The name including the full path of the executable file to launch.

-- CpuPriorityLevel (Citrix.Broker.Admin.SDK.CpuPriorityLevel)

The CPU priority of the launched process. Valid values are: Low, BelowNormal, Normal, AboveNormal, and High.

-- Description (System.String)

Optional application description. This description is visible to the end users.

-- Enabled (System.Boolean)

Specifies whether or not this application can be launched.

-- IconFromClient (System.Boolean)

Specifies if the app icon should be retrieved from the application on the client. This is reserved for possible future use, and all applications of type HostedOnDesktop cannot set or change this value.

-- IconUid (System.Int32?)

The icon UID used for this application. If not specified a generic icon is used.

-- MetadataKeys (System.String[])

All key names of metadata items associated with this application.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Metadata for this application.

-- Name (System.String)

Unique administrative name of application; this will include any parent admin folder hierarchy separated by backslash characters.

-- PublishedName (System.String)

Published name of application as seen by end user. If not specified value used defaults to the administrative name.

-- SecureCmdLineArgumentsEnabled (System.Boolean)

Specifies whether the command-line arguments should be secured.

-- ShortcutAddedToDesktop (System.Boolean)

Specifies whether a shortcut to the application should be placed on the user device.

-- ShortcutAddedToStartMenu (System.Boolean)

Specifies whether a shortcut to the application should be placed in the user's Start menu on their user device.

-- StartMenuFolder (System.String)

The name of the Start menu folder that holds the application shortcut.

-- Uid (System.Int32)

A unique identifier of an application.

-- UserFilterEnabled (System.Boolean)

Indicates if application-specific user filter is enabled.

-- UUID (System.Guid)

UUID of the application.

-- Visible (System.Boolean)

Specifies if the application is visible to users.

-- WaitForPrinterCreation (System.Boolean)

Specifies whether the VDA delays starting the app until printers are set up or not.

-- WorkingDirectory (System.String)

The working directory the executable is launched from.

Related topics

[New-BrokerApplication](#)

[Add-BrokerApplication](#)

[Remove-BrokerApplication](#)

[Rename-BrokerApplication](#)

[Move-BrokerApplication](#)

[Set-BrokerApplication](#)

Parameters

-Uid<Int32>

Gets only the application with the specified unique identifier.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Name<String>

Gets only the applications matching the specified name (including any parent admin folder hierarchy).

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminFolderName<String>

Gets applications that are in admin folders matching the specified name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminFolderUid<Int32>

Gets applications that are in the specified admin folder.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ApplicationName<String>

Gets applications that match the specified simple name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ApplicationType<ApplicationType>

Gets applications that match the type specified: HostedOnDesktop or InstalledOnClient.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedDesktopGroupPriority<Int32>

Gets applications with an associated desktop group identified by priority assigned to the pairing between an application and desktop group.

Associated desktop group is a desktop group on which the application is published.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedDesktopGroupUid<Int32>

Gets applications with an associated desktop group identified by the desktop group UID.

Associated desktop group is a desktop group on which the application is published.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedDesktopGroupUUID<Guid>

Gets applications with an associated desktop group identified by the desktop group UUID.

Associated desktop group is a desktop group on which the application is published.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedUserFullName<String>

Gets applications with an associated user identified by their full name (usually 'first-name last-name').

If the 'UserFilterEnabled' property is true then access to the application is restricted to those users only, otherwise access is unrestricted (but always subject to other policy rules).

Required?	false

Default Value	
Accept Pipeline Input?	false

-AssociatedUserName<String>

Gets applications with an associated user identified by their user name (in the form 'domain\user'). If the 'UserFilterEnabled' property is true then access to the application is restricted to those users only, otherwise access is unrestricted (but always subject to other policy rules).

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedUserUPN<String>

Gets applications with an associated user identified by their user principle name (in the form 'user@domain'). If the 'UserFilterEnabled' property is true then access to the application is restricted to those users only, otherwise access is unrestricted (but always subject to other policy rules).

Required?	false
Default Value	
Accept Pipeline Input?	false

-BrowserName<String>

Gets only the applications that match the supplied name. The BrowserName is usually an internal name for the application and is unique in the site.

Required?	false
Default Value	
Accept Pipeline Input?	false

-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

-ShortcutAddedToStart Menu<Boolean>

Gets only the applications that match depending on whether a shortcut for the application has been added to Start Menu of the user device or not.

Required?	false
Default Value	
Accept Pipeline Input?	false

-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

May 04, 2016

Gets the running applications on the desktops.

Syntax

```
Get-BrokerApplicationInstance -Uid <Int64> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerApplicationInstance [-ApplicationName <String>] [-ApplicationUid <Int32>] [-ApplicationUUID <Guid>] [-Instances  
<Int32>] [-MachineName <String>] [-MachineUid <Int32>] [-Metadata <String>] [-SessionKey <Guid>] [-SessionUid <Int64>] [-  
UserName <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter  
<String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Get-BrokerApplicationInstance gets the published applications that are running on desktops.

Only published applications that are launched from a Citrix client are returned. If a user launches an application from within a session (by double-clicking on an attachment from an email, for example) this will not show up in the list of running applications.

Also note that this is a list of launched published applications, not a list of processes running on the desktop. In some cases the original process associated with the published application may no longer be running, but if the session is still running the published application may be listed as running.

The number of instances for each published application running in a session is also returned. For example, if a user launches two Notepad applications from a Citrix client, and session-sharing occurs such that both Notepad applications run in the same session, then the Instances property indicates that 2 copies are running in the session.

See Get-BrokerApplication and Get-BrokerSession to get the details for the applications and sessions, respectively.

The Get-BrokerMachine cmdlet also returns a list of published applications that are running on a desktop. See the "ApplicationsInUse" attribute of the returned desktop objects.

----- BrokerApplicationInstance Object

The BrokerApplicationInstance object represents an instance of a published application in the site. It contains the following properties:

-- ApplicationName (System.String)

The administrative name of the application.

-- ApplicationUid (System.Int32)

The UID of the application.

-- ApplicationUUID (System.Guid)

The UUID of the application.

-- Instances (System.Int32)

The number of times this published application is running in the specified session.

-- MachineName (System.String)

Machine's SAM name (of the form domain\machine). If SAM name is unavailable, contains the machines's SID.

-- MachineUid (System.Int32)

UID of underlying machine.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Metadata for this application instance.

-- SessionKey (System.Guid)

The key of the session.

-- SessionUid (System.Int64)

The UID of the session.

-- Uid (System.Int64)

The unique identifier for this application instance object itself, distinct from the Uids of either application or session objects.

-- UserName (System.String)

User name (SAMName).

Related topics

[Get-BrokerApplication](#)

[Get-BrokerDesktop](#)

[Get-BrokerSession](#)

Parameters

-Uid<Int64>

Gets only the application instances specified by the unique identifier. This is the unique identifier for the application instance object itself, and is distinct from the Uids of either application or session objects.

Required?	true
Default Value	
Accept Pipeline Input?	false

-ApplicationName<String>

Gets only application instances for the specified application name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ApplicationUid<Int32>

Gets only application instances for the specified application Uid.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ApplicationUUID<Guid>

Gets only the application instances for the specified application UUID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Instances<Int32>

Gets only application instances that match the specified number of instances.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MachineName<String>

Gets only application instances running on the specified machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MachineUid<Int32>

Gets only application instances running on the machine with the specified UID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Metadata<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionKey<Guid>

Gets only application instances for the published applications running in the specified session.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionUid<Int64>

Gets only application instances for the published applications running in the specified session.

Required?	false
Default Value	
Accept Pipeline Input?	false

-UserName<String>

Gets only application instances being run by the specified users.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about_Broker_Filtering](#) for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell style filter expression. See about_Broker_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	

Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	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

May 04, 2016

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

May 04, 2016

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 user changes are persisted on machines in the catalog. Possible values are:

o OnLocal: User changes are stored on the machine's local storage.

o Discard: User changes are discarded.

o OnPvd: User changes are stored on the user's personal vDisk.

-- ProvisioningSchemeId (System.Guid?)

The GUID of the provisioning scheme (if any) associated with the catalog. This only applies if the provisioning type is MCS.

-- ProvisioningType (Citrix.Broker.Admin.SDK.ProvisioningType)

Specifies how the machines are provisioned in the catalog. Possible values are:

o Manual: No provisioning.

o PVS: Machine provisioned by PVS (machine may be physical, blade, VM...)

o MCS: Machine provisioned by MCS (machine must be a VM).

-- PvsAddress (System.String)

IP address of the PVS server to be used in a catalog with a PVS ProvisioningType.

-- PvsDomain (System.String)

The domain of the PVS server to be used in a catalog with a PVS ProvisioningType.

-- RemotePCDesktopGroupPriorities (System.Int32[])

Remote PC desktop groups' association priorities.

-- RemotePCDesktopGroupUids (System.Int32[])

UIDs of the Remote PC desktop groups associated with this catalog.

-- RemotePCHypervisorConnectionUid (System.Int32?)

UID of the hypervisor connection used for powering on RemotePC machines in this catalog (only for catalogs with IsRemotePC set to true).

-- Scopes (Citrix.Broker.Admin.SDK.ScopeReference[])

The list of the delegated admin scopes to which the catalog belongs.

-- SessionSupport (Citrix.Broker.Admin.SDK.SessionSupport)

Specifies the session support of the machines in the catalog. Valid values are:

SingleSession, MultiSession.

-- Uid (System.Int32)

Uid of the catalog.

-- UnassignedCount (System.Int32)

The number of unassigned machines (machines not assigned to users).

-- UsedCount (System.Int32)

The number of machines in the catalog that are in a desktop group.

-- UUID (System.Guid)

The global ID of the catalog.

Related topics

[New-BrokerCatalog](#)

[Remove-BrokerCatalog](#)

[Rename-BrokerCatalog](#)

Set-BrokerCatalog

Parameters

-Uid<Int32>

Get catalogs with the specified UID.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Name<String>

Gets catalogs with the specified name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AllocationType<AllocationType>

Gets catalogs that are of the specified allocation type. Values can be:

- o Static - Machines in a catalog of this type are permanently assigned to a user.
- o Permanent - equivalent to 'Static'.
- o Random - Machines in a catalog of this type are picked at random and temporarily assigned to a user.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssignedCount<Int32>

Gets catalogs containing a specified number of assigned machines (machines that have been assigned to users).

This property is typically used with advanced filtering; see about_Broker_Filtering.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AvailableAssignedCount<Int32>

Gets catalogs containing a specified number of available machines (those not in any desktop group) that are also assigned to users.

This property is typically used with advanced filtering; see about_Broker_Filtering.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AvailableCount<Int32>

Gets catalogs containing a specified number of available machines (those not in any desktop group).

This property is typically used with advanced filtering; see about_Broker_Filtering.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AvailableUnassignedCount<Int32>

Gets catalogs containing a specified number of available machines (those not in any desktop group) that are not assigned to users.

This property is typically used with advanced filtering; see about_Broker_Filtering.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Description<String>

Gets catalogs with the specified description.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HypervisorConnectionUid<Int32>

Gets catalogs associated with the specified hypervisor connection.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IsRemotePC<Boolean>

Gets catalogs with the specified IsRemotePC value.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MachinesArePhysical<Boolean>

Specifies whether machines in the catalog can be power-managed by the Citrix Broker Service. Where the Citrix Broker Service cannot control the power state of the machine specify \$true, otherwise \$false. Can only be specified together with a provisioning type of Pvs or Manual, or if used with the legacy CatalogKind parameter only with Pvs or PvsPvd catalog kinds.

Required?	false
Default Value	

Accept Pipeline Input?	false

-Metadata<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

Required?	false
Default Value	
Accept Pipeline Input?	false

-MinimumFunctionalLevel<FunctionalLevel>

Gets catalogs with a specific MinimumFunctionalLevel.

Valid values are L5, L7, L7_6

Required?	false
Default Value	
Accept Pipeline Input?	false

-PersistUserChanges<PersistUserChanges>

Gets catalogs with the specified behavior when persisting changes made by the end user. Possible values are:

- o OnLocal - User changes are stored on the machine's local storage.
- o Discard - User changes are discarded.
- o OnPvd - User changes are stored on the user's personal vDisk.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ProvisioningSchemeld<Guid>

Gets catalogs associated with the specified provisioning scheme.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ProvisioningType<ProvisioningType>

Specifies the provisioning type for the catalog. Values can be:

- o Manual - No provisioning.
- o PVS - Machine provisioned by PVS (machine may be physical, blade, VM,...).
- o MCS - Machine provisioned by MCS (machine must be VM).

Required?	false
Default Value	
Accept Pipeline Input?	false

-PvsAddress<String>

Gets catalogs containing machines provided by the Provisioning Services server with the specified address.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PvsDomain<String>

Gets catalogs containing machines provided by the Provisioning Services server in the specified domain.

Required?	false
Default Value	
Accept Pipeline Input?	false

-RemotePCDesktopGroupPriority<Int32>

Gets Remote PC catalogs with a Remote PC desktop group association with the specified priority.

Required?	false
Default Value	
Accept Pipeline Input?	false

-RemotePCDesktopGroupUid<Int32>

Gets Remote PC catalogs associated with the specified Remote PC desktop group.

Required?	false
Default Value	
Accept Pipeline Input?	false

-RemotePCHypervisorConnectionUid<Int32>

Gets Remote PC catalogs associated with the specified Remote PC hypervisor connection.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ScopeId<Guid>

Gets catalogs that are associated with the given scope identifier.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ScopeName<String>

Gets catalogs that are associated with the given scope name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionSupport<SessionSupport>

Gets catalogs that have the specified session capability. Values can be:

- o SingleSession - Single-session only machine.
- o MultiSession - Multi-session capable machine.

Required?	false
Default Value	
Accept Pipeline Input?	false

-UnassignedCount<Int32>

Gets catalogs containing a specified number of unassigned machines (machines not assigned to users).

This property is typically used with advanced filtering; see about_Broker_Filtering.

Required?	false
Default Value	
Accept Pipeline Input?	false

-UsedCount<Int32>

Gets catalogs containing a specified number of machines used in a desktop group.

This property is typically used with advanced filtering; see about_Broker_Filtering.

Required?	false

Default Value	
Accept Pipeline Input?	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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Gets Controllers running broker services in the site.

Syntax

```
Get-BrokerController [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerController [-MachineName <String>] [-ControllerVersion <String>] [-DesktopsRegistered <Int32>] [-DNSName <String>] [-LastActivityTime <DateTime>] [-LastLicensingServerEvent <LicensingServerEvent>] [-LastLicensingServerEventTime <DateTime>] [-LastStartTime <DateTime>] [-LicensingGraceState <LicensingGraceState>] [-LicensingServerState <LicensingServerState>] [-Metadata <String>] [-OSType <String>] [-OSVersion <String>] [-SID <String>] [-State <ControllerState>] [-UUID <Guid>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Gets Controllers from the current site that match the specified search criteria.

Controllers are server machines running instances of the broker service. The broker service is responsible for the brokering of user sessions to desktops or applications, and for power management of the underlying machines. An operational site must contain at least one Controller.

If no search criteria are specified, all Controllers in the site are obtained.

----- BrokerController Object

The BrokerController object represents a single controller running an instance of the Broker service. It contains the following properties:

-- ActiveSiteServices (System.String[])

The Broker site services active on the controller.

-- AssociatedHypervisorConnectionUids (System.Int32[])

The UIDs of the hypervisor connections being managed by the Broker service on the controller.

-- ControllerVersion (System.String)

The version of the Broker service on the controller.

-- DesktopsRegistered (System.Int32)

The number of VDA machines registered with the Broker service on the controller.

-- DNSName (System.String)

The DNS name of the controller.

-- LastActivityTime (System.DateTime?)

The last reported activity time of the Broker service on the controller.

-- LastLicensingServerEvent (Citrix.Broker.Admin.SDK.LicensingServerEvent?)

Last significant licensing server event reported by the Broker service on the controller.

-- LastLicensingServerEventDetails (System.String[])

Additional details associated with the last significant licensing server event.

-- LastLicensingServerEventTime (System.DateTime?)

Time at which the last significant licensing server event was reported.

-- LastStartTime (System.DateTime?)

The last start-up time of the Broker service on the controller.

-- LicensingGracePeriodReasons (Citrix.Broker.Admin.SDK.LicensingGracePeriodReason[])

Current active or expired licensing grace periods in effect on the controller.

-- LicensingGracePeriodTimesRemaining (System.TimeSpan[])

Times remaining in currently active or expired licensing grace periods in effect on the controller. Expired grace periods are indicated by zero remaining time. The number and order of entries in this list matches that in the LicensingGracePeriodReasons list.

-- LicensingGraceState (Citrix.Broker.Admin.SDK.LicensingGraceState)

The licensing grace state currently in effect in the Broker service on the controller.

-- LicensingServerState (Citrix.Broker.Admin.SDK.LicensingServerState)

The licensing server state currently in effect in the Broker service on the controller.

-- MachineName (System.String)

The Windows name of the controller.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

The metadata for the controller.

-- OSType (System.String)

The Operating System type of the controller.

-- OSVersion (System.String)

The Operating System version of the controller.

-- SID (System.String)

The SID of the controller.

-- State (Citrix.Broker.Admin.SDK.ControllerState)

The state of the Broker service on the controller.

-- Uid (System.Int32)

The UID of the controller instance.

-- UUID (System.Guid)

A globally unique identifier of the controller instance.

Related topics

[Get-BrokerDesktop](#)

Parameters

-Uid<Int32>

Gets only Controller with the specified unique ID.

Required?	true
Default Value	
Accept Pipeline Input?	false

-MachineName<String>

Gets only Controllers with the specified Windows name. ('domain\machine')

Required?	false
Default Value	
Accept Pipeline Input?	false

-ControllerVersion<String>

Gets only Controllers running the specified version of the broker service.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopsRegistered<Int32>

Gets only Controllers that have the specified number of desktops currently registered. This parameter is mainly of use with advanced filtering; see about_Broker_Filtering.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DNSName<String>

Gets only Controllers with the specified DNS name ('machine.domain')

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastActivityTime<DateTime>

Gets only Controllers last reported as active at the specified time. This parameter is mainly of use with advanced filtering; see about_Broker_Filtering.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastLicensingServerEvent<LicensingServerEvent>

Gets only Controllers with the specified last license server event recorded.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastLicensingServerEventTime<DateTime>

Gets only Controllers with its last recorded licensing server event at the specified time. This parameter is mainly of use with advanced filtering; see about_Broker_Filtering.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastStartTime<DateTime>

Gets only Controllers that last started-up at the specified time. This parameter is mainly of use with advanced filtering; see about_Broker_Filtering.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LicensingGraceState<LicensingGraceState>

Gets only Controllers in the specified licensing grace state.

Valid values are: NotActive, InOutOfBoxGracePeriod, InSupplementalGracePeriod, InEmergencyGracePeriod and GracePeriodExpired.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LicensingServerState<LicensingServerState>

Gets only Controllers in the specified licensing server state. Valid values are: ServerNotSpecified, NotConnected, OK, LicenseNotInstalled, LicenseExpired, Incompatible and Failed.

Required?	false

Default Value	
Accept Pipeline Input?	false

-Metadata<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

Required?	false
Default Value	
Accept Pipeline Input?	false

-OSType<String>

Gets only Controllers running the specified Operating System type.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OSVersion<String>

Gets only Controllers running the specified Operating System version.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SID<String>

Gets only Controllers with the specified SID.

Required?	false

Default Value	
Accept Pipeline Input?	false

-State<ControllerState>

Gets only Controllers currently in the specified state.

Valid values are: Failed, Off, On, and Active.

Required?	false
Default Value	
Accept Pipeline Input?	false

-UUID<Guid>

Gets only the Controller with the specified GUID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about_Broker_Filtering](#) for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by - ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell style filter expression. See about_Broker_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.Controller

Returns Controllers matching all specified selection criteria.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerController -State Active
```

Gets all Controllers in the site that are currently active (powered on and fully operational).

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerController -Filter 'LastStartTime -gt "-30:00"'
```

Gets all Controllers in the site that started-up in the last 30 minutes.

Get-BrokerDBConnection

May 04, 2016

Gets the database connection string for the specified data store used by the Broker Service.

Syntax

```
Get-BrokerDBConnection [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Gets the database connection string from the currently selected Broker Service instance.

If the returned string is blank, no valid connection string has been specified. In this case the service is running, but is idle and awaiting specification of a valid connection string.

The current service instance is the one on the local machine, or the one most recently specified using the -AdminAddress parameter of a Broker SDK cmdlet.

Related topics

[Set-BrokerDBConnection](#)

[Get-BrokerServiceStatus](#)

[Test-BrokerDBConnection](#)

Parameters

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

System.String

The database connection string configured for the current Broker Service instance.

Notes

If the command fails, the following errors can be returned.

Error Codes

NoDBConnections

The database connection string for the Broker Service has not been specified.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Get-BrokerDBConnection -AdminAddress controller1.mydomain.net
```

Gets the database connection string in use by the Broker Service instance running on controller "controller1.mydomain.net".

Get-BrokerDBSchema

May 04, 2016

Gets SQL scripts to create or maintain the database schema for the Citrix Broker Service.

Syntax

```
Get-BrokerDBSchema -DatabaseName <String> [-ServiceGroupName <String>] [-ScriptType <DatabaseScriptType>] [-SID <String>] [-LocalDatabase] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Gets SQL scripts that can be used to create a new Citrix Broker Service database schema, add a new Broker service to an existing site, remove a Broker service from a site, or create a database server logon for a Broker service.

If no Sid parameter is provided, the scripts obtained relate to the currently selected Broker service instance, otherwise the scripts relate to Broker service instance running on the machine identified by the Sid provided. When obtaining the Evict script, a Sid parameter must be supplied.

The current service instance is the one on the local machine, or the one most recently specified using the -AdminAddress parameter of a Broker SDK cmdlet.

The service instance used to obtain the scripts does not need to be a member of a site or to have had its database connection configured.

The database scripts support only Microsoft SQL Server, or SQL Server Express, and require Windows integrated authentication to be used. They can be run using SQL Server's SQLCMD utility, or by copying the script into an SQL Server Management Studio (SSMS) query window and executing the query. If using SSMS, the query must be executed in 'SQLCMD mode'.

The ScriptType parameter determines which script is obtained. If ScriptType is not specified, or is FullDatabase, the script contains:

- o Creation of service schema
- o Creation of database server logon
- o Creation of database user
- o Addition of database user to Broker service roles

If ScriptType is Instance, the returned script contains:

- o Creation of database server logon
- o Creation of database user
- o Addition of database user to Broker service roles

If ScriptType is Evict, the returned script contains:

- o Removal of Broker service instance from database

o Removal of database user

If ScriptType is Login, the returned script contains:

o Creation of database server logon only

If the service uses two data stores they can exist in the same database.

You do not need to configure a database before using this command.

Related topics

[Set-BrokerDBConnection](#)

[Test-BrokerDBConnection](#)

Parameters

-DatabaseName<String>

Specifies the name of the database into which the new Broker service schema is to be placed, or in which it already exists. The database itself is not created by any of the script types; it must already exist before the scripts are run.

Required?	true
Default Value	
Accept Pipeline Input?	false

-ServiceGroupName<String>

Specifies the name of the service group to be used when creating the database schema. The service group is a collection of all the Broker services that share the same database instance and are considered equivalent; that is, all the services within a service group can be used interchangeably.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ScriptType<DatabaseScriptType>

Specifies the type of database script returned. Available script types are:

-- FullDatabase

Creates a database schema for the Citrix Broker Service in a database instance that does not already contain one. This is

used when creating a new site. DatabaseName and ServiceGroupName are required parameters for this script type.

-- Instance

Adds a Broker Service instance to a database and so to the associated site. Appropriate database server logons and users are created to allow the service instance access to the required service schemas.

-- Evict

Removes a Broker Service instance from the database and so from the site. All reference to the service instance is removed from the database. DatabaseName and Sid are required parameters for this script type.

-- Login

Adds a logon for the Broker Service instance to a database server. This is specifically for use when configuring SQL Server mirroring where the mirror server must have appropriate logons created for all service instances in the site.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SID<String>

Specifies the SID of the controller on which the Broker Service instance to remove from the database is running (only valid for a script type of Evict).

Required?	false
Default Value	None
Accept Pipeline Input?	false

-LocalDatabase<SwitchParameter>

Specifies whether the database script is to be used in a database instance run on the same controller as other services in the service group. Including this parameter ensures the script creates only the required permissions for local services to access the database schema for Broker services. If this parameter is specified inappropriately, the service instance will not be able to connect to the database.

Required?	false
Default Value	

Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

System.String

A string containing the required SQL script for applying to a database.

Notes

The scripts returned support Microsoft SQL Server Express Edition, Microsoft SQL Server Standard Edition, and Microsoft SQL Server Enterprise Edition databases only, and are generated on the assumption that integrated authentication will be used.

If the ScriptType parameter is not included or set to 'FullDatabase', the full database script is returned, which will:

Create the database schema.

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist).

If the ScriptType parameter is set to 'Instance', the script will:

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist) and associate it with a user.

If the ScriptType parameter is set to 'Login', the script will:

Create the logon (providing the schema does not already exist) and associate it with a pre-existing user of the same name.

If the LocalDatabase parameter is included, the NetworkService account will be added to the list of accounts permitted to access the database. This is required only if the database is run on a controller.

If the command fails, the following errors can be returned.

Error Codes

GetSchemasFailed

The database schema could not be found.

ActiveDirectoryAccountResolutionFailed

The specified Active Directory account or Group could not be found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

```
C:\PS>Get-BrokerDBSchema -DatabaseName MySiteDB -ServiceGroupName MyServiceGroup > C:\BrokerSchema.sql
```

Gets a script to create the full database schema for the Citrix Broker Service and copies it to a file called "C:\BrokerSchema.sql"

This script can be used to create the service schema in a database with name "MySiteDB", which must already exist, and must not already contain a Broker service schema.

----- EXAMPLE 2 -----

```
C:\PS>Get-BrokerDBSchema -DatabaseName MySiteDB -ScriptType Login > C:\BrokerLogins.sql
```

Gets a script to create the appropriate database server logon for the Broker service. This can be used when configuring a mirror server for use.

Get-BrokerDBVersionChangeScript

May 04, 2016

Gets an SQL service schema update script for the Citrix Broker Service.

Syntax

```
Get-BrokerDBVersionChangeScript -DatabaseName <String> -TargetVersion <Version> [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

Gets an SQL script that can be used to update the current Citrix Broker Service database schema. An update can be an upgrade or downgrade.

A script can be obtained to update the current service schema to any version that is reachable by applying available schema update packages that have been uploaded by the Citrix Broker Service.

Typically, this mechanism is used to update the current service schema to a newer version, however it can also be used to revert previously applied updates.

The SQL script obtained can be run using SQL Server's SQLCMD utility, or by copying the script into an SQL Server Management Studio (SSMS) query window and executing the query. If using SSMS, the query must be executed in 'SQLCMD mode'.

The schema update packages used to generate update scripts are stored in the database; because of this, any Citrix Broker Service in the site can be used to obtain a schema update script.

The fact that an update package is available in the database does not mean that the update has actually been applied to the service's schema. In addition, application of an update does not remove the associated update packages.

Take care when using the update scripts. Citrix recommends that where possible service schema upgrades are performed using Studio rather than manually via the SDK. The database should be backed-up before an update is attempted. The database script may also require exclusive use of the schema, in which case all Citrix Broker Services must be shutdown before applying the update.

Once an update has been applied to the service schema, any existing Citrix Broker Services that are incompatible with the updated schema will cease to operate. The service state, as reported by `Get-BrokerServiceStatus`, provides information about the service compatibility (e.g. `DBNewerVersionThanService`).

Related topics

[Get-BrokerInstalledDBVersion](#)

[Get-BrokerServiceStatus](#)

[Get-BrokerDBSchema](#)

Parameters

-DatabaseName<String>

The name of the database containing the Citrix Broker Service schema to be updated.

Required?	true
Default Value	
Accept Pipeline Input?	false

-TargetVersion<Version>

The required target service schema version of the update. This is the service schema version obtained after the update script is applied.

Required?	true
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

System.Management.Automation.PSObject

The Get-BrokerDBVersionChangeScript cmdlet returns a PSObject containing a script that can be used to update the Citrix Broker Service database schema. The object has the following properties:

-- Script

The raw text of the SQL script to apply the update.

-- CanUndo

If true, indicates that after the update has been applied, a script to revert from the updated schema to the schema state prior to the update can be obtained. Because Get-BrokerDBVersionChangeScript gets only update scripts relating to the current schema version, a script to revert an update can be obtained only after the update has been applied.

-- NeedExclusiveAccess

If true, indicates that the update requires exclusive access to the Citrix Broker Service's schema while the update is applied; all Citrix Broker Services must be shutdown during the update.

Notes

The PSObject returned by this cmdlet contains the following properties:

-- Script The raw text of the SQL script to apply the update, or null in the case when no upgrade path to the specified target version exists.

-- NeedExclusiveAccess Indicates whether all services in the service group must be shut down during the update or not.

-- CanUndo Indicates whether the generated script allows the updated schema to be reverted to the state prior to the update.

Scripts to update the schema version are stored in the database so any service in the service group can obtain these scripts. Extreme caution should be exercised when using update scripts. Citrix recommends backing up the database before attempting to upgrade the schema. Database update scripts may require exclusive use of the schema and so may not be able to execute while any Broker services are running. However, this depends on the specific update being carried out.

After a schema update has been carried out, services that require the previous version of the schema may cease to operate. The ServiceState parameter reported by the Get-BrokerServiceStatus command provides information about service compatibility. For example, if the schema has been upgraded to a more recent version that a service cannot use, the service reports "DBNewerVersionThanService".

If the command fails, the following errors can be returned.

Error Codes

NoOp

The operation was successful but had no effect.

NoDBConnections

The database connection string for the Broker Service has not been specified.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

```
C:\PS> $update = Get-BrokerDBVersionChangeScript -DatabaseName MyDb -TargetVersion 1.0.75.0
```

```
C:\PS> $update.Script > update_75.sql
```

Gets an SQL update script to update the current schema to version 1.0.75.0. The resulting update_75.sql script is suitable for direct use with the SQL Server SQLCMD utility.

Get-BrokerDelayedHostingPowerAction

May 04, 2016

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

May 04, 2016

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[])

Usetnames of the users that have been associated with the desktop (usually in the form "domain\user").

Associated users are the current user(s) for shared desktops and the assigned users for private desktops.

-- AssociatedUserUPNs (System.String[])

The user principal names of the users that have been associated with the desktop (in the form user@upndomain.com).

Associated users are the current user(s) for shared desktops and the assigned users for private desktops.

-- AutonomouslyBrokered (System.Boolean?)

Session property indicating if the current session is an HDX session established by direct connection without being brokered.

Session properties are always null for multi-session desktops.

-- CatalogName (System.String)

Name of the catalog the desktop is a member of.

-- CatalogUid (System.Int32)

UID of the catalog the desktop is a member of.

-- ClientAddress (System.String)

Session property indicating the IP address of the client connected to the desktop.

Session properties are always null for multi-session desktops.

-- ClientName (System.String)

Session property indicating the host name of the client connected to the desktop.

Session properties are always null for multi-session desktops.

-- ClientVersion (System.String)

Session property indicating the version of the Citrix Receiver running on the connected client.

Session properties are always null for multi-session desktops.

-- ColorDepth (Citrix.Broker.Admin.SDK.ColorDepth?)

The color depth setting configured on the desktop, possible values are:

\$null, FourBit, EightBit, SixteenBit, and TwentyFourBit.

-- ConnectedViaHostName (System.String)

Session property indicating the host name of the connection gateway, router or client.

Session properties are always null for multi-session desktops.

-- ConnectedViaIP (System.String)

Session property indicating the IP address of the connection gateway, router or client.

Session properties are always null for multi-session desktops.

-- ControllerDNSName (System.String)

The DNS host name of the controller that the desktop is registered to.

-- DeliveryType (Citrix.Broker.Admin.SDK.DeliveryType)

Denotes whether the desktop delivers desktops only, apps only or both.

-- Description (System.String)

Description of the desktop.

-- DesktopConditions (System.String[])

List of outstanding desktop conditions for the desktop.

-- DesktopGroupName (System.String)

Name of the desktop group the desktop has been assigned to.

-- DesktopGroupUid (System.Int32)

Uid of the desktop group the desktop has been assigned to.

-- DesktopKind (Citrix.Broker.Admin.SDK.DesktopKind)

Deprecated.

Denotes whether the desktop is private or shared. AllocationType should be used instead.

-- DeviceId (System.String)

Session property indicating a unique identifier for the client device that has most recently been associated with the current session.

Session properties are always null for multi-session desktops.

-- DNSName (System.String)

The DNS host name of the desktop.

-- FunctionalLevel (Citrix.Broker.Admin.SDK.FunctionalLevel?)

The functional level of the desktop, if known.

-- HardwareId (System.String)

Session property indicating a unique identifier for the client hardware that has been most recently associated with the current session.

Session properties are always null for multi-session desktops.

-- HostedMachineId (System.String)

Unique ID within the hosting unit of the target managed desktop.

-- HostedMachineName (System.String)

The friendly name of a hosted desktop as used by its hypervisor. This is not necessarily the DNS name of the desktop.

-- HostingServerName (System.String)

DNS name of the hypervisor that is hosting the desktop if managed.

-- HypervisorConnectionName (System.String)

The name of the hypervisor connection that the desktop's hosting server is accessed through, if managed.

-- HypervisorConnectionUid (System.Int32?)

The UID of the hypervisor connection that the desktop's hosting server is accessed through, if managed.

-- IconUid (System.Int32?)

The UID of the desktop's icon that is displayed in StoreFront.

-- ImageOutOfDate (System.Boolean?)

Denotes whether the VM image for a hosted desktop is out of date.

-- InMaintenanceMode (System.Boolean)

Denotes whether the desktop is in maintenance mode.

-- IPAddress (System.String)

The IP address of the desktop.

-- IsAssigned (System.Boolean)

Denotes whether a private desktop has been assigned to a user/users, or a client name/address. Users can be assigned explicitly or by assigning on first use of the desktop.

-- IsPhysical (System.Boolean)

This value is true if the desktop is physical (ie not power managed by the Citrix Broker Service), and false otherwise.

-- LastConnectionFailure (Citrix.Broker.Admin.SDK.ConnectionFailureReason)

The reason for the last failed connection between a client and the desktop.

-- LastConnectionTime (System.DateTime?)

Time of the last detected connection attempt that either failed or succeeded.

-- LastConnectionUser (System.String)

The SAM name (in the form DOMAIN\user) of the user that last attempted a connection with the desktop. If the SAM name is not available, the SID is used.

-- LastDeregistrationReason (Citrix.Broker.Admin.SDK.DeregistrationReason?)

The reason for the last deregistration of the desktop with the broker. Possible values are:

AgentShutdown, AgentSuspended, AgentRequested, IncompatibleVersion, AgentAddressResolutionFailed, AgentNotContactable, AgentWrongActiveDirectoryOU, EmptyRegistrationRequest, MissingRegistrationCapabilities, MissingAgentVersion, InconsistentRegistrationCapabilities, NotLicensedForFeature, UnsupportedCredentialSecurityVersion, InvalidRegistrationRequest, SingleMultiSessionMismatch, FunctionalLevelTooLowForCatalog, FunctionalLevelTooLowForDesktopGroup, PowerOff, DesktopRestart, DesktopRemoved, AgentRejectedSettingsUpdate, SendSettingsFailure, SessionAuditFailure, SessionPrepareFailure, ContactLost, SettingsCreationFailure, UnknownError and BrokerRegistrationLimitReached.

-- LastDeregistrationTime (System.DateTime?)

Time of the last deregistration of the desktop from the controller.

-- LastErrorReason (System.String)

The reason for the last error detected in the desktop.

-- LastErrorTime (System.DateTime?)

The time of the last detected error.

-- LastHostingUpdateTime (System.DateTime?)

Time of last update to any hosting data (such as power state) for this desktop reported by the hypervisor connection.

-- LaunchedViaHostName (System.String)

Session property that denotes the host name of the StoreFront server used to launch the current brokered session.

Session properties are always null for multi-session desktops.

-- LaunchedViaIP (System.String)

Session property that denotes the IP address of the StoreFront server used to launch the current brokered session.

Session properties are always null for multi-session desktops.

-- MachineInternalState (Citrix.Broker.Admin.SDK.MachineInternalState)

The internal state of the machine associated with the desktop; reported while the desktop is registered to a controller, plus some private Citrix Broker Service states while the machine is not registered.

-- MachineName (System.String)

DNS host name of the machine associated with the desktop.

-- MachineUid (System.Int32)

Uid of the associated machine.

-- OSType (System.String)

A string that can be used to identify the operating system that is running on the desktop.

-- OSVersion (System.String)

A string that can be used to identify the version of the operating system running on the desktop, if known

-- PersistUserChanges (Citrix.Broker.Admin.SDK.PersistUserChanges)

Describes whether/how the user changes are persisted. Possible values are:

o OnLocal - Persist the user changes on the local disk of the desktop.

o Discard - Discard user changes.

o OnPvd - Persist user changes on the Citrix Personal vDisk.

-- PowerActionPending (System.Boolean)

Property indicating whether there are any pending power actions for the desktop.

-- PowerState (Citrix.Broker.Admin.SDK.PowerState)

The current power state of the desktop. Possible values are: Unmanaged, Unknown, Unavailable, Off, On, Suspended, TurningOn, TurningOff, Suspending, resuming.

-- Protocol (System.String)

Session property that denotes the protocol that the current session is using, can be either HDX, RDP or Console. Console sessions on XenDesktop 5 VDAs appear with a blank protocol.

Session properties are always null for multi-session desktops.

-- ProvisioningType (Citrix.Broker.Admin.SDK.ProvisioningType)

Describes how the machine associated with the desktop was provisioned, possible values are:

o Manual: No automated provisioning.

o PVS: Machine provisioned by PVS (may be physical, blade, VM,...)

o MCS: Machine provisioned by MCS (machine must be VM)

-- PublishedApplications (System.String[])

List of applications published by the desktop (displayed as browser names).

-- PublishedName (System.String)

The name of the desktop that is displayed in StoreFront, if the desktop is published.

-- PvdStage (Citrix.Broker.Admin.SDK.PvdStage)

For a desktop supporting Personal vDisk technology (PvD), indicates the stage of the PvD image preparation.

-- RegistrationState (Citrix.Broker.Admin.SDK.RegistrationState)

Indicates the registration state of the desktop. Possible values are: Unregistered, Initializing, Registered, 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 according to whether their current session is autonomously brokered or not. Autonomously brokered sessions are HDX sessions established by direct connection without being brokered.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-CatalogName<String>

Gets desktops from the catalog with the specific name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-CatalogUid<Int32>

Gets desktops from a catalog with a specific UID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ClientAddress<String>

Gets desktops with a specific client IP address.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ClientName<String>

Gets desktops with a specific client name.

Required?	false
Default Value	

Accept Pipeline Input?	false

-ClientVersion<String>

Gets desktops with a specific client version.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ColorDepth<ColorDepth>

Gets desktops configured with a specific color depth.

Valid values are FourBit, EightBit, SixteenBit, and TwentyFourBit.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ConnectedViaHostName<String>

Gets desktops with a specific host name of the incoming connection. This is usually a proxy or Citrix Access Gateway server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ConnectedViaIP<String>

Gets desktops with a specific IP address of the incoming connection.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ControllerDNSName<String>

Gets desktops with a specific DNS name of the controller they are registered with.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DeliveryType<DeliveryType>

Gets desktops of a particular delivery type.

Valid values are AppsOnly, DesktopsOnly, DesktopsAndApps

Required?	false
Default Value	
Accept Pipeline Input?	false

-Description<String>

Gets desktops with a specific description.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopCondition<String>

Gets desktop with an outstanding desktop condition condition.

Valid values are:

- o CPU: Indicates the machine has high CPU usage
- o ICALatency: Indicates the network latency is high
- o UPMLogonTime: Indicates that the profile load time was high

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopGroupName<String>

Gets desktops from a desktop group with the specified name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopGroupUid<Int32>

Gets desktops from a desktop group with the specified UID.

Required?	false

Default Value	
Accept Pipeline Input?	false

-DesktopKind<DesktopKind>

Deprecated: Use AllocationType parameter.

Gets desktops of a particular kind.

Valid values are Private, Shared.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DeviceId<String>

Gets desktops with a specific client device ID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DNSName<String>

Gets desktops with a specific DNS name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-FunctionalLevel<FunctionalLevel>

Gets desktops with a specific FunctionalLevel.

Valid values are L5, L7, L7_6

Required?	false
Default Value	
Accept Pipeline Input?	false

-HardwareId<String>

Gets desktops with a specific client hardware ID.

Required?	false

Default Value	
Accept Pipeline Input?	false

-HostedMachineId<String>

Gets desktops with a specific machine ID known to the hypervisor.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HostedMachineName<String>

Gets desktops with a specific machine name known to the hypervisor.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HostingServerName<String>

Gets desktops with a specific name of the hosting hypervisor server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HypervisorConnectionName<String>

Gets desktops with a specific name of the hosting hypervisor connection.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HypervisorConnectionUid<Int32>

Gets desktops with a specific UID of the hosting hypervisor connection.

Required?	false
Default Value	

Accept Pipeline Input?	false

-IconUid<Int32>

Gets desktops with a specific configured icon. Note that desktops with a null IconUid use the icon of the desktop group.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ImageOutOfDate<Boolean>

Gets desktops by whether their disk image is out of date (for machines provisioned using MCS only).

Required?	false
Default Value	
Accept Pipeline Input?	false

-InMaintenanceMode<Boolean>

Gets desktops with a specific InMaintenanceMode setting.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IPAddress<String>

Gets desktops with a specific IP address.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IsAssigned<Boolean>

Gets desktops according to whether they are assigned or not. Desktops may be assigned to one or more users or groups, a client IP address or a client endpoint name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IsPhysical<Boolean>

Specifies if machines in the catalog can be power managed by the Citrix Broker Service. Where the power state of the machine cannot be controlled, specify \$true, otherwise \$false. Can only be specified together with a provisioning type of Pvs or Manual, or if used with the deprecated CatalogKind parameter only with Pvs or PvsPvd catalog kinds.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastConnectionFailure<ConnectionFailureReason>

Gets desktops with a specific reason for the last recorded connection failure. This value is None if the last connection was successful or if there has been no attempt to connect to the desktop yet.

Valid values are None, SessionPreparation, RegistrationTimeout, ConnectionTimeout, Licensing, Ticketing, and Other.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastConnectionTime<DateTime>

Gets desktops that last connected at a specific time. This is the time that the broker detected that the connection attempt either succeeded or failed.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastConnectionUser<String>

Gets desktops where a specific user name last attempted a connection (in the form 'domain\user').

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastDeregistrationReason<DeregistrationReason>

Gets desktops whose broker last recorded a specific deregistration reason.

Valid values are \$null, AgentShutdown, AgentSuspended, AgentRequested, IncompatibleVersion, AgentAddressResolutionFailed, AgentNotContactable, AgentWrongActiveDirectoryOU, EmptyRegistrationRequest, MissingRegistrationCapabilities, MissingAgentVersion, InconsistentRegistrationCapabilities, NotLicensedForFeature, UnsupportedCredentialSecurityVersion, InvalidRegistrationRequest, SingleMultiSessionMismatch, FunctionalLevelTooLowForCatalog, FunctionalLevelTooLowForDesktopGroup, PowerOff, DesktopRestart, DesktopRemoved, AgentRejectedSettingsUpdate, SendSettingsFailure, SessionAuditFailure, SessionPrepareFailure, ContactLost, SettingsCreationFailure, UnknownError and BrokerRegistrationLimitReached.

Required?	false

Default Value	
Accept Pipeline Input?	false

-LastDeregistrationTime<DateTime>

Gets desktops by the time that they were last deregistered.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastErrorReason<String>

Gets desktops with the specified last error reason.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastErrorTime<DateTime>

Gets desktops with the specified last error time.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastHostingUpdateTime<DateTime>

Gets desktops with a specific time that the hosting information was last updated.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LaunchedViaHostName<String>

Gets desktops with a specific host name of the StoreFront server from which the user launched the session.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	

Accept Pipeline Input?	false

-LaunchedViaIP<String>

Gets desktops with a specific IP address of the StoreFront server from which the user launched the session.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MachineInternalState<MachineInternalState>

Gets desktops with the specified internal machine state.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MachineUid<Int32>

Gets desktops with a specific machine UID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OSType<String>

Gets desktops by the type of operating system they are running.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OSVersion<String>

Gets desktops by the version of the operating system they are running.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PersistUserChanges<PersistUserChanges>

Gets desktops by the location where the user changes are persisted.

- o OnLocal - User changes are persisted locally.
- o Discard - User changes are discarded.
- o OnPvd - User changes are persisted on the Pvd.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PowerActionPending<Boolean>

Gets desktops with a specific power action pending state.

Valid values are \$true or \$false.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PowerState<PowerState>

Gets desktops with a specific power state.

Valid values are Unmanaged, Unknown, Unavailable, Off, On, Suspended, TurningOn, TurningOff, Suspending, and Resuming.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Protocol<String>

Gets desktops with connections using a specific protocol, for example HDX, RDP, or Console.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ProvisioningType<ProvisioningType>

Gets desktops that are in a catalog with a particular provisioning type. Values can be:

- o Manual - No provisioning.

o PVS - Machine provisioned by PVS (machine may be physical, blade, VM,...).

o MCS - Machine provisioned by MCS (machine must be VM).

Required?	false
Default Value	
Accept Pipeline Input?	false

-PublishedApplication<String>

Gets desktops with a specific application published to them.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PublishedName<String>

Gets desktops with a specific published name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PvdStage<PvdStage>

Gets desktops with a specific personal vDisk stage.

Valid values are None, Requested, Starting, Working and Failed.

Required?	false
Default Value	
Accept Pipeline Input?	false

-RegistrationState<RegistrationState>

Gets desktops with a specific registration state.

Valid values are Unregistered, Initializing, Registered and AgentError.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SecureIcaActive<Boolean>

Gets desktops depending on whether the current session uses SecureICA or not.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SecureIcaRequired<Boolean>

Gets desktops configured with a particular SecureIcaRequired setting. Note that the desktop setting of \$null indicates that the desktop group value is used.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionHidden<Boolean>

Gets desktops by whether their sessions are hidden or not. Hidden sessions are treated as though they do not exist when launching sessions; a hidden session cannot be reconnected to, but a new session may be launched using the same entitlement.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionId<Int32>

Deprecated.

Gets desktops by session ID, a unique identifier that Remote Desktop Services uses to track the session but it is only unique on that machine.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionState<SessionState>

Gets desktops with a specific session state.

Valid values are \$null, Other, PreparingSession, Connected, Active, Disconnected, Reconnecting, NonBrokeredSession, and Unknown.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionStateChangeTime<DateTime>

Gets desktops whose sessions last changed state at a specific time.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionUid<Int64>

Gets single-session desktops with a specific session UID (\$null for no session).

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionUserName<String>

Gets desktops with a specific user name for the current session (in the form 'domain\user').

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionUserSID<String>

Gets desktops with a specific SID of the current session user.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SID<String>

Gets desktops with a specific machine SID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SmartAccessTag<String>

Gets session desktops where the session has the specific SmartAccess tag.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-StartTime<DateTime>

Gets desktops with a specific session start time.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SummaryState<DesktopSummaryState>

Gets desktops with a specific summary state.

Valid values are Off, Unregistered, Available, Disconnected, InUse and Preparing.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Tag<String>

Gets desktops with a specific tag.

Required?	false
Default Value	
Accept Pipeline Input?	false

-WillShutDownAfterUse<Boolean>

Gets desktops depending on whether they shut down after use or not.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ApplicationUid<Int32>

Gets desktops with a specific published application (identified by its UID).

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about_Broker_Filtering for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each

name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell style filter expression. See about_Broker_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.Desktop

Get-BrokerDesktop returns an object for each matching desktop.

Notes

To compare dates or times, use -Filter and relative comparisons. For more information, see about_Broker_Filtering and the examples.

Examples

----- **EXAMPLE 1** -----

C:\PS> Get-BrokerDesktop -RegistrationState Unregistered

```
C:\PS> Get-BrokerDesktop -Filter { RegistrationState -ne 'Registered' }
```

Both commands retrieve desktops that are unregistered. The second command also includes desktops with a registration state of AgentError.

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerDesktop -SessionUid $null | ft -a DNSName,SummaryState
```

Gets desktops without sessions, listing the DNS name and current state.

----- **EXAMPLE 3** -----

```
C:\PS> Get-BrokerDesktop -Filter { OSType -like "Windows XP*" -and ImageOutOfDate }
```

Finds all Windows XP desktops with an out-of-date image.

----- **EXAMPLE 4** -----

```
C:\PS> Get-BrokerDesktop -ApplicationInUse '*powerpoint'
```

Gets desktops running a published PowerPoint application. It matches any application browser name containing the word 'powerpoint'. String comparisons are case-insensitive.

----- **EXAMPLE 5** -----

```
C:\PS> Get-BrokerDesktop -DesktopCondition * DNSName,SessionUserName,DesktopConditions
```

Finds all desktops with an outstanding desktop condition, listing the affected desktop and user.

----- **EXAMPLE 6** -----

```
C:\PS> $d = (Get-Date).AddDays(-1)
```

```
C:\PS> Get-BrokerDesktop -Filter { StartTime -le $d } | ft MachineName,SessionUserName,StartTime,@{Label='Duration'; Expression=((Get-Date) - $_.StartTime)}
```

Finds users who have been logged on for more than a day, and outputs the machine name, start time, and duration the session has been logged on.

Get-BrokerDesktopGroup

May 04, 2016

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>] [-SettlementPeriodBeforeAutoShutdown <TimeSpan>] [-ShutdownDesktopsAfterUse <Boolean>] [-Tag <String>] [-TimeZone <String>] [-TotalApplications <Int32>] [-TurnOnAddedMachine <Boolean>] [-UUID <Guid>] [-ApplicationUid <Int32>] [-TagUid <Int32>] [-PowerTimeSchemeUid <Int32>] [-MachineConfigurationUid <Int32>] [-RemotePCCatalogUid <Int32>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Retrieve desktop groups matching the specified criteria. If no parameters are specified this cmdlet enumerates all desktop groups.

Desktop groups represent groups of desktops that are managed together for brokering purposes.

----- BrokerDesktopGroup Object

A desktop group object represents a collection of machines that are fully configured in a site that is able to run either a Microsoft Windows desktop environment, individual applications, or both.

-- AutomaticPowerOnForAssigned (System.Boolean)

Specifies whether assigned desktops in the desktop group are automatically started at the start of peak time periods. Only relevant for groups whose DesktopKind is Private.

-- AutomaticPowerOnForAssignedDuringPeak (System.Boolean)

Specifies whether assigned desktops in the desktop are automatically started throughout peak time periods. Only relevant for groups whose DesktopKind is Private and which have AutomaticPowerOnForAssigned set to true.

-- ColorDepth (Citrix.Broker.Admin.SDK.ColorDepth)

Default color depth of sessions started with machines in the desktop group. Possible values are:

FourBit, EightBit, SixteenBit, TwentyFourBit.

-- ConfigurationSlotUids (System.Int32[])

Uids of any configuration slots which hold machine configurations associated with the desktop group. The order of slot UIDs in this list correspond with the order of items in the associated MachineConfigurationNames and MachineConfigurationUids list properties, and so the same slot UID can appear more than once.

-- DeliveryType (Citrix.Broker.Admin.SDK.DeliveryType)

The type of resources being published. Possible values are:

DesktopsOnly, AppsOnly, DesktopsAndApps.

-- Description (System.String)

Description of the desktop group.

-- DesktopKind (Citrix.Broker.Admin.SDK.DesktopKind)

The kind of the desktops being published, possible values are:

Private and Shared.

-- DesktopsAvailable (System.Int32)

The number of machines in the desktop group in state Available; this is the number of machines with no sessions present.

-- DesktopsDisconnected (System.Int32)

The number of disconnected sessions present on machines in the desktop group.

-- DesktopsInUse (System.Int32)

The number of machines in the desktop group in state InUse; this is the number of machines with at least one session present.

-- DesktopsNeverRegistered (System.Int32)

The number of machines in the desktop group that have never registered with the current site.

-- DesktopsPreparing (System.Int32)

The number of machines in the desktop group whose PvD disk image is being prepared.

-- DesktopsUnregistered (System.Int32)

The number of machines in the desktop group that are currently unregistered.

-- Enabled (System.Boolean)

Specifies whether the desktop group is enabled or not; disabled desktop groups do not appear to users.

-- IconUid (System.Int32)

The Uid of the icon to be used as a default for desktops in the desktop group. Individual desktop objects can override this default by setting the IconUid parameter on the desktop object.

-- InMaintenanceMode (System.Boolean)

Specifies whether the machines in the desktop group are in maintenance mode or not.

-- IsRemotePC (System.Boolean)

Specifies whether the desktop group is a Remote PC desktop group.

-- MachineConfigurationNames (System.String[])

The MachineConfiguration names associated with the desktop group.

-- MachineConfigurationUids (System.Int32[])

The MachineConfiguration uids associated with the desktop group.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Metadata associated with the desktop group.

-- MinimumFunctionalLevel (Citrix.Broker.Admin.SDK.FunctionalLevel)

The minimum FunctionalLevel required for the machines in the desktop group to be able to register with the Citrix Broker Service.

-- Name (System.String)

Name of the desktop group.

-- OffPeakBufferSizePercent (System.Int32)

The percentage of machines that are kept available in an idle state outside peak hours.

-- OffPeakDisconnectAction (Citrix.Broker.Admin.SDK.SessionChangeHostingAction)

The action that is performed after a configurable period of a user session disconnecting outside peak hours. Possible values are Nothing, Suspend or Shutdown.

-- OffPeakDisconnectTimeout (System.Int32)

The number of minutes before the configured action is performed after a user session disconnects outside peak hours.

-- OffPeakExtendedDisconnectAction (Citrix.Broker.Admin.SDK.SessionChangeHostingAction)

The action performed after a second configurable period of a user session disconnecting outside peak hours. Possible values are Nothing, Suspend, or Shutdown.

-- OffPeakExtendedDisconnectTimeout (System.Int32)

The number of minutes before the second configured action is performed after a user session disconnects outside peak hours.

-- OffPeakLogOffAction (Citrix.Broker.Admin.SDK.SessionChangeHostingAction)

The action performed after a configurable period of a user session ending outside peak hours. Possible values are Nothing, Suspend, or Shutdown.

-- OffPeakLogOffTimeout (System.Int32)

The number of minutes before the configured action is performed after a user session ends outside peak hours.

-- PeakBufferSizePercent (System.Int32)

The percentage of machines in the desktop group that are kept available in an idle state in peak hours.

-- PeakDisconnectAction (Citrix.Broker.Admin.SDK.SessionChangeHostingAction)

The action performed after a configurable period of a user session disconnecting in peak hours. Possible values are Nothing, Suspend, or Shutdown.

-- PeakDisconnectTimeout (System.Int32)

The number of minutes before the configured action is performed after a user session disconnects in peak hours.

-- PeakExtendedDisconnectAction (Citrix.Broker.Admin.SDK.SessionChangeHostingAction)

The action performed after a second configurable period of a user session disconnecting in peak hours. Possible values are Nothing, Suspend, or Shutdown.

-- PeakExtendedDisconnectTimeout (System.Int32)

The number of minutes before the second configured action is performed after a user session disconnects in peak hours.

-- PeakLogOffAction (Citrix.Broker.Admin.SDK.SessionChangeHostingAction)

The action performed after a configurable period of a user session ending in peak hours. Possible values are Nothing, Suspend, or Shutdown.

-- PeakLogOffTimeout (System.Int32)

The number of minutes before the configured action is performed after a user session ends in peak hours.

-- ProtocolPriority (System.String[])

A list of protocol names in the order in which they are attempted for use during connection.

-- PublishedName (System.String)

The name of the desktop group as it is to appear to the user in StoreFront.

-- Scopes (Citrix.Broker.Admin.SDK.ScopeReference[])

The list of the delegated admin scopes to which the desktop group belongs.

-- SecureIcaRequired (System.Boolean)

Flag that specifies if the SecureICA encryption of the HDX protocol is required for sessions of desktops in the desktop

group.

-- Sessions (System.Int32)

The total number of user sessions currently running on all of the machines in the desktop group.

-- SessionSupport (Citrix.Broker.Admin.SDK.SessionSupport)

Specifies the session support (single/multi) of the machines in the desktop group. Machines with the incorrect session support for the desktop group will be unable to register with the Citrix Broker Service.

-- SettlementPeriodBeforeAutoShutdown (System.TimeSpan)

Time after a session ends during which automatic shutdown requests (for example, shutdown after use, idle pool management) are deferred. Any outstanding shutdown request takes effect after the settlement period expires. This is typically used to configure time to allow logoff scripts to complete.

-- ShutdownDesktopsAfterUse (System.Boolean)

Specifies if the desktops will shut down after they have been used and there are no sessions running on the machine. The machines will not shut down if they are placed into maintenance mode, even if this flag is set to \$true. The machines, however, will shutdown after the machine is taken out of maintenance mode if the flag is still set.

-- Tags (System.String[])

Tags associated with the desktop group.

-- TimeZone (System.String)

The timezone that desktops in the desktop group are in (for power policy purposes).

-- TotalApplications (System.Int32)

Total number of applications associated with the desktop group.

-- TotalDesktops (System.Int32)

Total number of machines in the desktop group.

-- TurnOnAddedMachine (System.Boolean)

Specifies whether the broker should attempt to turn on power-managed machines when they are added to the desktop group.

-- Uid (System.Int32)

Uid of the desktop group.

-- UUID (System.Guid)

UUID of the desktop group.

Related topics

[New-BrokerDesktopGroup](#)

[Set-BrokerDesktopGroup](#)

[Rename-BrokerDesktopGroup](#)

[Remove-BrokerDesktopGroup](#)

[Add-BrokerUser](#)

[Add-BrokerTag](#)

Parameters

-Uid<Int32>

Gets desktop groups with the specified value of Uid.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Name<String>

Gets desktop groups whose name matches the supplied pattern.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AutomaticPowerOnForAssigned<Boolean>

Gets only desktop groups with the specified value of AutomaticPowerOnForAssigned.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AutomaticPowerOnForAssignedDuringPeak<Boolean>

Gets only desktop groups with the specified value of AutomaticPowerOnForAssignedDuringPeak.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ColorDepth<ColorDepth>

Gets only desktop groups with the specified color depth.

Valid values are FourBit, EightBit, SixteenBit, and TwentyFourBit.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DeliveryType<DeliveryType>

Gets desktop groups according to their delivery type.

Valid values are DesktopsOnly, AppsOnly and DesktopsAndApps.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Description<String>

Gets desktop groups whose description matches the supplied pattern.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopKind<DesktopKind>

Gets desktops of a particular kind.

Valid values are Private and Shared.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Enabled<Boolean>

Gets desktop groups with the specified value of Enabled.

Required?	false
Default Value	
Accept Pipeline Input?	false

-InMaintenanceMode<Boolean>

Gets desktop groups with the specified value of InMaintenanceMode.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IsRemotePC<Boolean>

Gets desktop groups with the specified IsRemotePC value.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Metadata<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

Required?	false
Default Value	
Accept Pipeline Input?	false

-MinimumFunctionalLevel<FunctionalLevel>

Gets desktop groups with a specific MinimumFunctionalLevel.

Valid values are L5, L7, L7_6

Required?	false
Default Value	
Accept Pipeline Input?	false

-OffPeakBufferSizePercent<Int32>

Gets desktop groups with the specified value of OffPeakBufferSizePercent.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OffPeakDisconnectAction<SessionChangeHostingAction>

Gets desktop groups with the specified value of OffPeakDisconnectAction.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OffPeakDisconnectTimeout<Int32>

Gets desktop groups with the specified value of OffPeakDisconnectTimeout.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OffPeakExtendedDisconnectAction<SessionChangeHostingAction>

Gets desktop groups with the specified value of OffPeakExtendedDisconnectAction.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OffPeakExtendedDisconnectTimeout<Int32>

Gets desktop groups with the specified value of OffPeakExtendedDisconnectTimeout.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OffPeakLogOffAction<SessionChangeHostingAction>

Gets desktop groups with the specified value of OffPeakLogOffAction.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OffPeakLogOffTimeout<Int32>

Gets desktop groups with the specified value of OffPeakLogOffTimeout.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PeakBufferSizePercent<Int32>

Gets desktop groups with the specified value of PeakBufferSizePercent.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PeakDisconnectAction<SessionChangeHostingAction>

Gets desktop groups with the specified value of PeakDisconnectAction.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PeakDisconnectTimeout<Int32>

Gets desktop groups with the specified value of PeakDisconnectTimeout.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PeakExtendedDisconnectAction<SessionChangeHostingAction>

Gets desktop groups with the specified value of PeakExtendedDisconnectAction.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PeakExtendedDisconnectTimeout<Int32>

Gets desktop groups with the specified value of PeakExtendedDisconnectTimeout.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PeakLogOffAction<SessionChangeHostingAction>

Gets desktop groups with the specified value of PeakLogOffAction.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PeakLogOffTimeout<Int32>

Gets desktop groups with the specified value of PeakLogOffTimeout.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PublishedName<String>

Gets desktop groups whose published name matches the supplied pattern.

Required?	false

Default Value	
Accept Pipeline Input?	false

-ScopeId<Guid>

Gets desktop groups that are associated with the given scope identifier.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ScopeName<String>

Gets desktop groups that are associated with the given scope name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SecureIcaRequired<Boolean>

Gets desktop groups with the specified value of SecureIcaRequired.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionSupport<SessionSupport>

Gets desktop groups that have the specified session capability. Values can be:

- o SingleSession - Single-session only machine.
- o MultiSession - Multi-session capable machine.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SettlementPeriodBeforeAutoShutdown<TimeSpan>

Gets desktop groups with the specified value of SettlementPeriodBeforeAutoShutdown.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ShutdownDesktopsAfterUse<Boolean>

Gets desktop groups with the specified value of ShutdownDesktopsAfterUse.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Tag<String>

Gets desktop groups tagged with the specified tag.

Required?	false
Default Value	
Accept Pipeline Input?	false

-TimeZone<String>

Gets desktop groups with the specified value of TimeZone.

Required?	false

Default Value	
Accept Pipeline Input?	false

-TotalApplications<Int32>

Gets desktop groups that are acting as delivery groups for the specified number of applications.

Required?	false
Default Value	
Accept Pipeline Input?	false

-TurnOnAddedMachine<Boolean>

Gets desktop groups with the specified value of TurnOnAddedMachine value.

Required?	false
Default Value	
Accept Pipeline Input?	false

-UUID<Guid>

Gets desktop groups with the specified value of UUID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ApplicationUid<Int32>

Gets desktop groups that publish the specified application (identified by Uid)

Required?	false
Default Value	

Accept Pipeline Input?	false

-TagUid<Int32>

Gets desktop groups to which the specified tag (identified by its Uid) has been added to help identify it - see Add-BrokerTag for more information.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PowerTimeSchemeUid<Int32>

Gets desktop groups associated with the specified power time scheme (identified by its Uid).

Required?	false
Default Value	
Accept Pipeline Input?	false

-MachineConfigurationUid<Int32>

Gets desktop groups with the specified value of MachineConfiguration.

Required?	false
Default Value	
Accept Pipeline Input?	false

-RemotePCCatalogUid<Int32>

Gets Remote PC desktop groups associated with the specified catalog.

Required?	false
Default Value	

Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about_Broker_Filtering for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-Sort By<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false

Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell style filter expression. See [about_Broker_Filtering](#) for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.DesktopGroup

Get-BrokerDesktopGroup returns an object for each matching desktop group.

Notes

To perform greater-than or less-than comparisons, use -Filter. For more information, see [about_Broker_Filtering](#) and the examples.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerDesktopGroup -PublishedName EMEA*
```

Finds all desktop groups with published names starting with "EMEA".

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerDesktopGroup -InMaintenanceMode $true
```

Finds all desktop groups in maintenance mode.

Get-BrokerDesktopUsage

May 04, 2016
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

May 04, 2016

Gets desktop rules from the site's entitlement policy.

Syntax

```
Get-BrokerEntitlementPolicyRule [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerEntitlementPolicyRule [[-Name] <String>] [-BrowserName <String>] [-ColorDepth <ColorDepth>] [-Description <String>] [-DesktopGroupUid <Int32>] [-Enabled <Boolean>] [-ExcludedUser <User>] [-ExcludedUserFilterEnabled <Boolean>] [-IconUid <Int32>] [-IncludedUser <User>] [-IncludedUserFilterEnabled <Boolean>] [-Metadata <String>] [-PublishedName <String>] [-SecureIcaRequired <Boolean>] [-SessionReconnection <SessionReconnection>] [-UUID <Guid>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Returns desktop rules matching the specified search criteria from the site's entitlement policy. If no search criteria are specified, all desktop rules in the entitlement policy are obtained.

A desktop rule in the entitlement policy defines the users who are allowed per-session access to a machine from the rule's associated desktop group to run a full desktop session.

----- BrokerEntitlementPolicyRule Object

The BrokerEntitlementPolicyRule object represents a single desktop rule within the site's entitlement policy. It contains the following properties:

-- BrowserName (System.String)

Site-wide unique name identifying this desktop entitlement to other components (for example StoreFront).

-- ColorDepth (Citrix.Broker.Admin.SDK.ColorDepth?)

The color depth of any desktop session launched by the user from the entitlement. If null, the equivalent setting from the rule's desktop group is used.

-- Description (System.String)

Optional description of the rule. The text may be visible to the end user, for example, as a tooltip associated with the desktop entitlement.

-- DesktopGroupUid (System.Int32)

The unique ID of the desktop group to which the rule applies.

-- Enabled (System.Boolean)

Indicates whether the rule is enabled. A disabled rule is ignored when evaluating the site's entitlement policy.

-- ExcludedUserFilterEnabled (System.Boolean)

Indicates whether the excluded users filter is enabled. If the filter is disabled then any user entries in the filter are ignored when entitlement policy rules are evaluated.

-- ExcludedUsers (Citrix.Broker.Admin.SDK.ChbUser[])

The excluded users filter of the rule, that is, the users and groups who are explicitly denied an entitlement to a desktop session from this rule.

-- IconUid (System.Int32?)

The unique ID of the icon used to display the desktop entitlement to the user. If null, the equivalent setting from the rule's desktop group is used.

-- IncludedUserFilterEnabled (System.Boolean)

Indicates whether the included users filter is enabled. If the filter is disabled then any user who satisfies the requirements of the access policy is implicitly granted an entitlement to a desktop session by the rule.

-- IncludedUsers (Citrix.Broker.Admin.SDK.ChbUser[])

The included users filter of the rule, that is, the users and groups who are granted an entitlement to a desktop session by the rule.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

A collection of arbitrary key/value pairs that can be associated with the rule. The administrator can use these values for any purpose; they are not used by the site itself in any way.

-- Name (System.String)

The administrative name of the rule. Each rule in the site's entitlement policy must have a unique name (irrespective of whether they are desktop or application rules).

-- PublishedName (System.String)

The name of the desktop session entitlement as seen by the user. If null, the equivalent setting from the rule's desktop group is used.

-- SecureIcaRequired (System.Boolean?)

Indicates whether the rule requires the SecureICA protocol for desktop sessions launched using the entitlement. If null, the equivalent setting from the rule's desktop group is used.

-- SessionReconnection (Citrix.Broker.Admin.SDK.SessionReconnection)

Defines reconnection (roaming) behavior for sessions launched using this rule. Session reconnection control is an experimental and unsupported feature.

-- Uid (System.Int32)

The unique ID of the rule itself.

-- UUID (System.Guid)

UUID of the rule.

Related topics

[New-BrokerEntitlementPolicyRule](#)

[Set-BrokerEntitlementPolicyRule](#)

[Rename-BrokerEntitlementPolicyRule](#)

[Remove-BrokerEntitlementPolicyRule](#)

Parameters

-Uid<Int32>

Gets the desktop rule with the specified unique ID.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Name<String>

Gets only desktop rules with the specified name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-BrowserName<String>

Gets only desktop rules with browser names matching the specified name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ColorDepth<ColorDepth>

Gets only desktop rules with the specified color depth.

Valid values are \$null, FourBit, EightBit, SixteenBit, and TwentyFourBit.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Description<String>

Gets only desktop rules with the specified description.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopGroupUid<Int32>

Gets only desktop rules that apply to the desktop group with the specified unique ID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Enabled<Boolean>

Gets only desktop rules that are in the specified state, either enabled (\$true), or disabled (\$false).

Required?	false
Default Value	
Accept Pipeline Input?	false

-ExcludedUser<User>

Gets only desktop rules that have the specified user in their excluded users filter (whether the filter is enabled or not).

Required?	false
Default Value	
Accept Pipeline Input?	false

-ExcludedUserFilterEnabled<Boolean>

Gets only desktop rules that have their excluded user filter enabled (\$true) or disabled (\$false).

Required?	false
Default Value	
Accept Pipeline Input?	false

-IconUid<Int32>

Gets only desktop rules using the icon with the specified unique ID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedUser<User>

Gets only desktop rules that have the specified user in their included users filter (whether the filter is enabled or not).

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedUserFilterEnabled<Boolean>

Gets only desktop rules that have their included user filter enabled (\$true) or disabled (\$false).

Required?	false
Default Value	
Accept Pipeline Input?	false

-Metadata<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

Required?	false
Default Value	
Accept Pipeline Input?	false

-PublishedName<String>

Gets only desktop rules with the specified published name, that is, the desktop session entitlement name that the end user sees.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SecureIcaRequired<Boolean>

Gets only desktop rules that require the desktop session to use the SecureICA protocol (\$true) or not (\$false).

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionReconnection<SessionReconnection>

Gets only desktop rules with the specified session reconnection behavior.

Required?	false
Default Value	
Accept Pipeline Input?	false

-UUID<Guid>

Gets rules with the specified value of UUID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about_Broker_Filtering](#) for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by - ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell style filter expression. See about_Broker_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.EntitlementPolicyRule

Get-BrokerEntitlementPolicyRule returns all desktop entitlement policy rules that match the specified selection criteria.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerEntitlementPolicyRule
```

Returns all desktop rules from the entitlement policy. This offers a complete description of the current site's entitlement policy with respect to desktops published from shared desktop groups.

----- **EXAMPLE 2** -----

```
C:\PS> $dg = Get-BrokerDesktopGroup 'Customer Support'
```

```
C:\PS> Get-BrokerEntitlementPolicyRule -DesktopGroupUid $dg.Uid
```

Returns all desktop rules in the entitlement policy that give users entitlements to desktop sessions in the Customer Support desktop group.

Get-BrokerHostingPowerAction

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Gets hypervisor connections matching the specified criteria.

Syntax

```
Get-BrokerHypervisorConnection [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerHypervisorConnection [-Name] <String> [-HypHypervisorConnectionUid <Guid>] [-IsReady <Boolean>] [-MachineCount <Int32>] [-MaxAbsoluteActiveActions <Int32>] [-MaxAbsoluteNewActionsPerMinute <Int32>] [-MaxAbsolutePvdPowerActions <Int32>] [-MaxPercentageActiveActions <Int32>] [-MaxPvdPowerActionsPercentageOfDesktops <Int32>] [-Metadata <String>] [-PreferredController <String>] [-State <HypervisorConnectionState>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Get-BrokerHypervisorConnection cmdlet gets hypervisor connections matching the specified criteria. If no parameters are specified this cmdlet enumerates all hypervisor connections.

----- BrokerHypervisorConnection Object

The BrokerHypervisorConnection represents hypervisor connection object. It contains the following properties:

-- Capabilities (System.String[])

The set of capabilities as reported by the hypervisor.

-- HypHypervisorConnectionUid (System.Guid)

The Guid that identifies the hypervisor connection.

-- IsReady (System.Boolean)

Indicates that the connection is ready to be used in the configuration of managed machines.

-- MachineCount (System.Int32)

Count of machines associated with this hypervisor connection.

-- MaxAbsoluteActiveActions (System.Int32?)

Maximum number of active power actions allowed at any one time (defined in the metadata named 'Citrix_Broker_MaxAbsoluteActiveActions' on the hypervisor connection in the Citrix Hosting Service).

-- MaxAbsoluteNewActionsPerMinute (System.Int32?)

Maximum number of new actions that can be fired off to the hypervisor in any one minute (defined in the metadata named 'Citrix_Broker_MaxAbsoluteNewActionsPerMinute' on the hypervisor connection in the Citrix Hosting Service).

-- MaxAbsolutePvdPowerActions (System.Int32?)

Maximum number of active Pvd power actions allowed at any one time (defined in the metadata named 'Citrix_Broker_MaxAbsolutePvdPowerActions' on the hypervisor connection in the Citrix Hosting Service).

-- MaxPercentageActiveActions (System.Int32?)

Maximum percentage of machines on the connection that can have active power actions at any one time (defined in the metadata named 'Citrix_Broker_MaxPowerActionsPercentageOfDesktops' on the hypervisor connection in the Citrix Hosting Service).

-- MaxPvdPowerActionsPercentageOfDesktops (System.Int32?)

Maximum percentage of machines on the connection that can be in personal VDisk image preparation mode at any one time (defined in the metadata named 'Citrix_Broker_MaxPvdPowerActionsPercentageOfDesktops' on the hypervisor connection in the Citrix Hosting Service).

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Collection of all the metadata associated to the hypervisor connection.

-- Name (System.String)

The display name of the hypervisor connection.

-- PreferredController (System.String)

The name of the controller which is preferred to be used, when available, to perform all communication to the hypervisor. The name is in DOMAIN\machine form. A preferred controller may have been automatically chosen when the hypervisor connection was created.

-- State (Citrix.Broker.Admin.SDK.HypervisorConnectionState)

The state of the hypervisor connection.

-- Uid (System.Int32)

Unique internal identifier of hypervisor connection.

Related topics

[New-BrokerHypervisorConnection](#)

[Remove-BrokerHypervisorConnection](#)

[Set-BrokerHypervisorConnection](#)

Parameters

-Uid<Int32>

Gets the hypervisor connection with the specified internal id.

Required?	true
Default Value	

Accept Pipeline Input?	false

-Name<String>

Gets hypervisor connections with the specified name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HypervisorConnectionUid<Guid>

Gets hypervisor connections with the specified Guid.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IsReady<Boolean>

Gets hypervisor connections with the specified value of the IsReady flag.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MachineCount<Int32>

Gets hypervisor connections with the specified machine count.

Required?	false
Default Value	

Accept Pipeline Input?	false

-MaxAbsoluteActiveActions<Int32>

Gets hypervisor connections with the specified MaxAbsoluteActiveActions value.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MaxAbsoluteNewActionsPerMinute<Int32>

Gets hypervisor connections with the specified MaxAbsoluteNewActionsPerMinute value.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MaxAbsolutePvdPowerActions<Int32>

Gets hypervisor connections with the specified MaxAbsolutePvdPowerActions value.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MaxPercentageActiveActions<Int32>

Gets hypervisor connections with the specified MaxPercentageActiveActions value.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MaxPvdPowerActionsPercentageOfDesktops<Int32>

Gets hypervisor connections with the specified MaxPvdPowerActionsPercentageOfDesktops value.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Metadata<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

Required?	false
Default Value	
Accept Pipeline Input?	false

-PreferredController<String>

Gets hypervisor connections with the specified preferred controller. Specify the SAM name of the controller.

Required?	false
Default Value	
Accept Pipeline Input?	false

-State<HypervisorConnectionState>

Gets hypervisor connections with the specified connection state. Values can be can be:

- o Unavailable - The broker is unable to contact the hypervisor.
- o InMaintenanceMode - The hosting server is in maintenance mode.
- o On - The broker is in contact with the hypervisor.

Required?	false

Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about_Broker_Filtering](#) for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if

no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell style filter expression. See [about_Broker_Filtering](#) for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through `Select-Object`, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None

Return Values

Citrix.Broker.Admin.SDK.HypervisorConnection

Get-BrokerHypervisorConnection returns an object for each matching hypervisor connection.

Examples

----- EXAMPLE 1 -----

```
c:\PS> $hvConn = Get-BrokerHypervisorConnection -Name "hvConnectionName"
```

Gets a hypervisor connection by name.

----- EXAMPLE 2 -----

```
c:\PS> $hvConn = Get-BrokerHypervisorConnection -PreferredController "domainName\controllerName"
```

Gets hypervisor connections by preferred controller.

----- EXAMPLE 3 -----

```
c:\PS> $machine = Get-BrokerMachine -Uid $machineUid
```

```
c:\PS> $hvConn = Get-BrokerHypervisorConnection -Uid $machine.HypervisorConnectionUid
```

Gets hypervisor connection used by a (power managed) machine.

Get-BrokerIcon

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Gets a list of all available database schema versions for the Broker Service.

Syntax

```
Get-BrokerInstalledDbVersion [-Upgrade] [-Downgrade] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Gets the current version number of the Citrix Broker Service database schema when called with no parameters.

When called with the -Upgrade parameter, gets the service schema version numbers to which an upgrade could be performed.

When called with the -Downgrade parameter, gets the service schema version numbers to which a downgrade could be performed.

The SQL scripts to perform schema upgrades and downgrades can be obtained using the Get-BrokerDBVersionChangeScript cmdlet. Citrix recommends that where possible service schema upgrades are performed using Studio rather than manually via the SDK.

Only one of the -Upgrade or -Downgrade parameters may be supplied at once.

Related topics

[Get-BrokerDBVersionChangeScript](#)

[Get-BrokerDBSchema](#)

Parameters

-Upgrade<SwitchParameter>

Specifies that only schema versions to which the current database version can be updated should be returned.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Downgrade<SwitchParameter>

Specifies that only schema versions to which the current database version can be reverted should be returned.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

System.Version

Get-BrokerInstalledDBVersion returns database schema version numbers as requested.

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

Both the Upgrade and Downgrade flags were specified.

NoOp

The operation was successful but had no effect.

NoDBConnections

The database connection string for the Broker Service has not been specified.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

```
C:\PS>Get-BrokerInstalledDBVersion
```

Gets the current Citrix Broker Service database schema version number.

----- EXAMPLE 2 -----

```
C:\PS>Get-BrokerInstalledDBVersion -Upgrade
```

Get the versions of the Broker Service database schema for which upgrade scripts are supplied.

Get-BrokerLease

May 04, 2016

Gets stored leases.

Syntax

```
Get-BrokerLease [-Uid] <Int64> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerLease [[-Key] <String>] [-Expiration <DateTime>] [-LastModified <DateTime>] [-LeaseType <BrokerLeaseType>] [-OwnerSAMName <String>] [-OwnerSID <String>] [-OwnerUPN <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Gets leases filtered by specific Uid or Owner information.

----- BrokerLease Object

The BrokerLease object represents a single instance of a lease. It contains the following properties:

-- Expiration (System.DateTime)

The expiration time of the lease.

-- Key (System.String)

The SHA1 representation of the lease key.

-- LastModified (System.DateTime)

The modification time of the lease.

-- LeaseType (Citrix.Broker.Admin.SDK.BrokerLeaseType)

The type of lease.

-- OwnerSAMName (System.String)

The SAM name of the user associated with the lease.

-- OwnerSID (System.String)

The SID of the user associated with the lease.

-- OwnerUPN (System.String)

The UPN of the user associated with the lease.

-- Uid (System.Int64)

The UID of the lease itself.

-- Value (System.String)

The serialized lease data in XML.

Related topics

[Remove-BrokerLease](#)

[Update-BrokerLocalLeaseCache](#)

Parameters

-Uid<Int64>

Gets only the lease specified by unique identifier.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Key<String>

Gets only the leases matching the specified lease key pattern.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Expiration<DateTime>

Gets only the leases matching the specified expiration date and time.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastModified<DateTime>

Gets only the leases matching the specified modified date and time.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LeaseType<BrokerLeaseType>

Gets only leases of a specific type. Possible values Enumeration, Launch.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OwnerSAMName<String>

Gets only the leases associated with the specified Domain\User.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OwnerSID<String>

Gets only the leases associated with the specified user SID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OwnerUPN<String>

Gets only the leases associated with the specified user UPN.

Required?	false

Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about_Broker_Filtering for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if

no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell style filter expression. See [about_Broker_Filtering](#) for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through `Select-Object`, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None Input cannot be piped to this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.Lease

Returns an Lease object for each enumerated lease.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> $lease = Get-BrokerLease -Uid 1
```

Gets the lease with internal Uid 1.

----- **EXAMPLE 2** -----

```
C:\PS> $leases = Get-BrokerLease -OwnerSAMName Domain\User
```

Gets the leases associated with the specified user.

Get-BrokerMachine

May 04, 2016

Gets machines belonging to this site.

Syntax

```
Get-BrokerMachine [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerMachine [-MachineName] <String> [-AgentVersion <String>] [-AllocationType <AllocationType>] [-ApplicationInUse <String>] [-AssignedClientName <String>] [-AssignedIPAddress <String>] [-AssociatedUserFullName <String>] [-AssociatedUserName <String>] [-AssociatedUserSID <String>] [-AssociatedUserUPN <String>] [-BrowserName <String>] [-CatalogName <String>] [-CatalogUid <Int32>] [-CatalogUUID <Guid>] [-ColorDepth <ColorDepth>] [-ControllerDNSName <String>] [-DeliveryType <DeliveryType>] [-Description <String>] [-DesktopCondition <String>] [-DesktopGroupName <String>] [-DesktopGroupUid <Int32>] [-DesktopGroupUUID <Guid>] [-DesktopKind <DesktopKind>] [-DesktopUid <Int32>] [-DNSName <String>] [-FaultState <MachineFaultState>] [-FunctionalLevel <FunctionalLevel>] [-HostedMachineId <String>] [-HostedMachineName <String>] [-HostingServerName <String>] [-HypervisorConnectionName <String>] [-HypervisorConnectionUid <Int32>] [-HypHypervisorConnectionUid <Guid>] [-IconUid <Int32>] [-ImageOutOfDate <Boolean>] [-InMaintenanceMode <Boolean>] [-IPAddress <String>] [-IsAssigned <Boolean>] [-IsPhysical <Boolean>] [-LastConnectionFailure <ConnectionFailureReason>] [-LastConnectionTime <DateTime>] [-LastConnectionUser <String>] [-LastDeregistrationReason <DeregistrationReason>] [-LastDeregistrationTime <DateTime>] [-LastErrorReason <String>] [-LastErrorTime <DateTime>] [-LastHostingUpdateTime <DateTime>] [-LastPvdErrorReason <String>] [-LastPvdErrorTime <DateTime>] [-LoadIndex <Int32>] [-MachineInternalState <MachineInternalState>] [-Metadata <String>] [-OSType <String>] [-OSVersion <String>] [-PersistUserChanges <PersistUserChanges>] [-PowerActionPending <Boolean>] [-PowerState <PowerState>] [-ProvisioningType <ProvisioningType>] [-PublishedApplication <String>] [-PublishedName <String>] [-PvdEstimatedCompletionTime <DateTime>] [-PvdPercentDone <Int32>] [-PvdStage <PvdStage>] [-PvdUpdateStartTime <DateTime>] [-RegistrationState <RegistrationState>] [-ScheduledReboot <ScheduledReboot>] [-SecureIcaRequired <Boolean>] [-SessionAutonomouslyBrokered <Boolean>] [-SessionClientAddress <String>] [-SessionClientName <String>] [-SessionClientVersion <String>] [-SessionConnectedViaHostName <String>] [-SessionConnectedViaIP <String>] [-SessionCount <Int32>] [-SessionDeviceId <String>] [-SessionHardwareId <String>] [-SessionHidden <Boolean>] [-SessionKey <Guid>] [-SessionLaunchedViaHostName <String>] [-SessionLaunchedViaIP <String>] [-SessionProtocol <String>] [-SessionSecureIcaActive <Boolean>] [-SessionsEstablished <Int32>] [-SessionSmartAccessTag <String>] [-SessionsPending <Int32>] [-SessionStartTime <DateTime>] [-SessionState <SessionState>] [-SessionStateChangeTime <DateTime>] [-SessionSupport <SessionSupport>] [-SessionType <SessionType>] [-SessionUid <Int64>] [-SessionUserName <String>] [-SessionUserSID <String>] [-SID <String>] [-SummaryState <DesktopSummaryState>] [-SupportedPowerActions <String[]>] [-Tag <String>] [-UUID <Guid>] [-VMToolsState <VMToolsState>] [-WillShutdownAfterUse <Boolean>] [-WindowsConnectionSetting <WindowsConnectionSetting>] [-AssignedUserSID <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Retrieves machines matching the specified criteria. If no parameters are specified, this cmdlet enumerates all machines.

Get-BrokerMachine returns objects that combine machine configuration and state information.

For single-session machines, session information is displayed if present. If "fast user switching" is enabled, more than one session may be present on single-session machines. Because this cmdlet returns information only for a single session, if two

sessions are present it will return information about the brokered session (rather than, for example, an unbrokered direct RDP session). If there is no session running, session-related fields return \$null.

For multi-session machines, no session information about single sessions is displayed by this cmdlet, and so are always \$null. Get-BrokerSession can be used to get information about sessions on both multi-session and single-session machines.

To count machines, rather than retrieve full details of each machine, use Group-BrokerMachine instead.

See about_Broker_Filtering for information about advanced filtering options, and about_Broker_Machines for background information about machines.

----- BrokerMachine Object

The machine object returned represents a physical or virtual machine, which has been configured in the site.

-- AgentVersion (System.String)

Version of the Citrix Virtual Delivery Agent (VDA) installed on the machine.

-- AllocationType (Citrix.Broker.Admin.SDK.AllocationType)

Describes how the machine is allocated to the user, can be Permanent or Random.

-- ApplicationsInUse (System.String[])

List of applications in use on the machine (in the form of browser name).

-- AssignedClientName (System.String)

The name of the endpoint client device that the machine has been assigned to.

-- AssignedIPAddress (System.String)

The IP address of the endpoint client device that the machine has been assigned to.

-- AssociatedUserFullNames (System.String[])

Full names of the users that have been associated with the machine (usually in the form "Firstname Lastname").

Associated users are the current user(s) for shared machines and the assigned users for private machines.

-- AssociatedUserNames (System.String[])

Usernames of the users that have been associated with the machine (in the form "domain\user").

Associated users are the current user(s) for shared machines and the assigned users for private machines.

-- AssociatedUserSIDs (System.String[])

The SIDs of the users that have been associated with the machine.

Associated users are the current user(s) for shared machines and the assigned users for private machines.

-- AssociatedUserUPNs (System.String[])

The User Principal Names of the users associated with the machine (in the form user@domain).

Associated users are the current user(s) for shared machines and the assigned users for private machines.

-- BrowserName (System.String)

Site-wide unique name identifying associated desktop to other components (for example StoreFront). This is typically non-null only for machines backing assigned private desktops.

-- Capabilities (System.String[])

List of the capabilities that the machine supports. Valid capabilities are:

- o MultiSession: Indicates an RDS- (Terminal Services-) based machine, which supports multiple active sessions from different users.

- o CBP1_5: Indicates the machine uses the CBP 1.5 protocol for communication.

-- CatalogName (System.String)

Name of the catalog the machine is a member of.

-- CatalogUid (System.Int32)

UID of the catalog the machine is a member of.

-- CatalogUUID (System.Guid)

UUID of the catalog the machine is a member of.

-- ColorDepth (Citrix.Broker.Admin.SDK.ColorDepth?)

The color depth setting configured on the machine, possible values are:

Snul, FourBit, EightBit, SixteenBit, and TwentyFourBit.

-- ControllerDNSName (System.String)

The DNS host name of the controller that the machine is registered to.

-- DeliveryType (Citrix.Broker.Admin.SDK.DeliveryType?)

Denotes whether the machine delivers desktops only, apps only or both.

-- Description (System.String)

Description of the machine.

-- DesktopConditions (System.String[])

List of outstanding desktop conditions for the machine.

-- DesktopGroupName (System.String)

Name of the desktop group the machine is a member of.

-- DesktopGroupUid (System.Int32?)

UID of the desktop group the machine is a member of.

-- DesktopGroupUUID (System.Guid?)

UUID of the desktop group the machine is a member of.

-- DesktopKind (Citrix.Broker.Admin.SDK.DesktopKind?)

Deprecated.

Denotes whether the machine is private or shared. Use AllocationType instead.

-- DesktopUid (System.Int32?)

The UID of the associated desktop object.

-- DNSName (System.String)

The DNS host name of the machine.

-- FaultState (Citrix.Broker.Admin.SDK.MachineFaultState)

Summary state of any current fault state of the machine. Can be one of the following:

o None - No fault; machine is healthy.

o FailedToStart - Last power-on operation for machine failed.

o StuckOnBoot - Machine does not seem to have booted following power on.

o Unregistered - Machine has failed to register within expected period, or its registration has been rejected.

o MaxCapacity - Machine is reporting itself at maximum capacity.

-- FunctionalLevel (Citrix.Broker.Admin.SDK.FunctionalLevel?)

Functional level of the machine, if known.

-- HostedMachineId (System.String)

Unique ID within the hosting unit of the target managed machine.

-- HostedMachineName (System.String)

The friendly name of a hosted machine as used by its hypervisor. This is not necessarily the DNS name of the machine.

-- HostingServerName (System.String)

DNS name of the hypervisor that is hosting the machine if managed.

-- HypervisorConnectionName (System.String)

The name of the hypervisor connection that the machine has been assigned to, if managed.

-- HypervisorConnectionUid (System.Int32?)

The UID of the hypervisor connection that the machine's hosting server is accessed through.

-- HypHypervisorConnectionUid (System.Guid?)

The UUID of the hypervisor connection that the machine's hosting server is accessed through

-- IconUid (System.Int32?)

The UID of the machine's icon that is displayed in StoreFront.

-- ImageOutOfDate (System.Boolean?)

Denotes if the VM image for a hosted machine is out of date.

-- InMaintenanceMode (System.Boolean)

Denotes if the machine is in maintenance mode.

-- IPAddress (System.String)

The IP address of the machine.

-- IsAssigned (System.Boolean)

Denotes whether a private desktop has been assigned to a user/users, or a client name/address. Users can be assigned explicitly or by assigning on first use of the machine.

-- IsPhysical (System.Boolean)

This value is true if the machine is physical (ie not power managed by the Citrix Broker service, and false otherwise.

-- LastConnectionFailure (Citrix.Broker.Admin.SDK.ConnectionFailureReason)

The reason for the last failed connection between a client and the machine.

-- LastConnectionTime (System.DateTime?)

Time of the last detected connection attempt that either failed or succeeded.

-- LastConnectionUser (System.String)

The SAM name (in the form DOMAIN\user) of the user that last attempted a connection with the machine. If the SAM name is not available, the SID is used.

-- LastDeregistrationReason (Citrix.Broker.Admin.SDK.DeregistrationReason?)

The reason for the last deregistration of the machine with the broker. Possible values are:

AgentShutdown, AgentSuspended, AgentRequested, IncompatibleVersion, AgentAddressResolutionFailed, AgentNotContactable, AgentWrongActiveDirectoryOU, EmptyRegistrationRequest, MissingRegistrationCapabilities, MissingAgentVersion, InconsistentRegistrationCapabilities, NotLicensedForFeature, UnsupportedCredentialSecurityVersion, InvalidRegistrationRequest, SingleMultiSessionMismatch, FunctionalLevelTooLowForCatalog, FunctionalLevelTooLowForDesktopGroup, PowerOff, DesktopRestart, DesktopRemoved, AgentRejectedSettingsUpdate, SendSettingsFailure, SessionAuditFailure, SessionPrepareFailure, ContactLost, SettingsCreationFailure, UnknownError and BrokerRegistrationLimitReached.

-- LastDeregistrationTime (System.DateTime?)

Time of the last deregistration of the machine from the controller.

-- LastErrorReason (System.String)

The reason for the last error detected in the machine.

-- LastErrorTime (System.DateTime?)

The time of the last detected error.

-- LastHostingUpdateTime (System.DateTime?)

Time of last update to any hosting data (such as power states) for this machine reported by the hypervisor connection.

-- LastPvdErrorReason (System.String)

The error text from the most recent failure of the Personal vDisk preparation process for this machine (if any).

-- LastPvdErrorTime (System.DateTime?)

The time of the most recent failure of the Personal vDisk preparation process for this machine (if any).

-- LoadIndex (System.Int32?)

Gives current effective load index for multi-session machines.

-- LoadIndexes (System.String[])

Gives the last reported individual load indexes that were used in the calculation of the LoadIndex value. Note that the LoadIndex value may have been subsequently adjusted due to session brokering operations. This value is only set for multi-session machines.

-- MachineInternalState (Citrix.Broker.Admin.SDK.MachineInternalState)

The internal state of the machine; reported while the machine is registered to a controller, plus some private Citrix Broker Service states while the machine is not registered.

-- MachineName (System.String)

DNS host name of the machine.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Any metadata that is associated with the machine.

-- OSType (System.String)

A string that can be used to identify the operating system that is running on the machine.

-- OSVersion (System.String)

A string that can be used to identify the version of the operating system running on the machine, if known.

-- PersistUserChanges (Citrix.Broker.Admin.SDK.PersistUserChanges)

Describes if and how user changes are persisted. Possible values are:

- o OnLocal - Persist the user changes on the local disk of the machine.

- o Discard - Discard user changes.

- o OnPvd - Persist user changes on the Citrix Personal vDisk.

-- PowerActionPending (System.Boolean)

Indicates if there are any pending power actions for the machine.

-- PowerState (Citrix.Broker.Admin.SDK.PowerState)

The current power state of the machine. Possible values are: Unmanaged, Unknown, Unavailable, Off, On, Suspended, TurningOn, TurningOff, Suspending, resuming.

-- ProvisioningType (Citrix.Broker.Admin.SDK.ProvisioningType)

Describes how the machine was provisioned, possible values are:

- o Manual: No automated provisioning.

- o PVS: Machine provisioned by PVS (may be physical, blade, VM,...)

- o MCS: Machine provisioned by MCS (machine must be VM)

-- PublishedApplications (System.String[])

List of applications published by the machine (displayed as browser names).

-- PublishedName (System.String)

The name of the machine that is displayed in StoreFront, if the machine has been published.

-- PvdEstimatedCompletionTime (System.DateTime?)

If preparation of the Personal vDisk is currently in progress for this machine, this reports an estimation of the time at which the process will be complete.

-- PvdPercentDone (System.Int32?)

If preparation of the Personal vDisk is currently in progress for this machine, this reports how far the process has got as a percentage. This value will be zero if preparation is not in progress.

-- PvdStage (Citrix.Broker.Admin.SDK.PvdStage)

For a machine supporting Personal vDisk technology (PvD), indicates the stage of the PvD image preparation.

-- PvdUpdateStartTime (System.DateTime?)

If preparation of the Personal vDisk is currently in progress for this machine, this reports when the update process began.

-- RegistrationState (Citrix.Broker.Admin.SDK.RegistrationState)

Indicates the registration state of the machine. Possible values are: Unregistered, Initializing, Registered, AgentError.

-- ScheduledReboot (Citrix.Broker.Admin.SDK.ScheduledReboot)

Indicates the state of any scheduled reboot operation for a machine. Possible values:

o None: No reboot is scheduled.

o Pending: Machine is awaiting reboot but is available for use.

o Draining: Machine is awaiting reboot and is unavailable for new sessions; reconnections to existing connections are still allowed, however.

o InProgress: Machine is actively undergoing a scheduled reboot. o Natural: Natural reboot in progress. Machine is awaiting a restart.

-- SecureIcaRequired (System.Boolean?)

Flag indicating whether SecureICA is required or not when starting a session on the machine.

-- SessionAutonomouslyBrokered (System.Boolean?)

Session property indicating if the current session is an HDX session established by direct connection without being brokered.

Session properties are always null for multi-session machines.

-- SessionClientAddress (System.String)

Session property indicating the IP address of the client connected to the machine.

Session properties are always null for multi-session machines.

-- SessionClientName (System.String)

Session property indicating the host name of the client connected to the machine.

Session properties are always null for multi-session machines.

-- SessionClientVersion (System.String)

Session property indicating the version of the Citrix Receiver on the connected client.

Session properties are always null for multi-session machines.

-- SessionConnectedViaHostName (System.String)

Session property indicating the host name of the connection gateway, router or client.

Session properties are always null for multi-session machines.

-- SessionConnectedViaIP (System.String)

Session property indicating the IP address of the connection gateway, router or client.

Session properties are always null for multi-session machines.

-- SessionCount (System.Int32)

Count of number of sessions on the machine.

-- SessionDeviceId (System.String)

Session property indicating a unique identifier for the client device that has most recently been associated with the current session.

Session properties are always null for multi-session machines.

-- SessionHardwareId (System.String)

Session property indicating a unique identifier for the client hardware that has been most recently associated with the current session.

Session properties are always null for multi-session machines.

-- SessionHidden (System.Boolean?)

Session property that indicates if a session is hidden.

Session properties are always null for multi-session machines.

-- SessionKey (System.Guid?)

Session property indicating the key of the current session.

Session properties are always null for multi-session machines.

-- SessionLaunchedViaHostName (System.String)

Session property that denotes the host name of the StoreFront server used to launch the current brokered session.

Session properties are always null for multi-session machines.

-- SessionLaunchedViaIP (System.String)

Session property that denotes the IP address of the StoreFront server used to launch the current brokered session.

Session properties are always null for multi-session machines.

-- SessionProtocol (System.String)

Session property that denotes the protocol that the current session is using, can be either HDX, RDP or Console. Console sessions on XenDesktop 5 VDAs appear with a blank protocol.

Session properties are always null for multi-session machines.

-- SessionSecureIcaActive (System.Boolean?)

Session property that indicates whether SecureICA is active on the current session or not.

Session properties are always null for multi-session machines.

-- SessionsEstablished (System.Int32)

Number of established sessions on this machine. For multi-session machines this excludes established sessions which have not yet completed their logon processing.

-- SessionSmartAccessTags (System.String[])

Session property that indicates the Smart Access tags for the current session.

Session properties are always null on multi-session machines.

-- SessionsPending (System.Int32)

Number of pending (brokered but not yet established) sessions on this machine. For multi-session machines this also includes established sessions which have not yet completed their logon processing.

-- SessionStartTime (System.DateTime?)

Session property that indicates the start time of the current session.

Session properties are always null on multi-session machines.

-- SessionState (Citrix.Broker.Admin.SDK.SessionState?)

Session property indicating the state of the current session, possible values are:

Other, PreparingSession, Connected, Active, Disconnected, Reconnecting, NonBrokeredSession and Unknown. Session properties are always null for multi-session machines.

-- SessionStateChangeTime (System.DateTime?)

Session property indicating the time of the last state change of the current session.

Session properties are always null for multi-session machines.

-- SessionSupport (Citrix.Broker.Admin.SDK.SessionSupport)

Indicates the session support of the machine.

Possible values:

o SingleSession: Single-session only machine.

o MultiSession: Multi-session capable machine.

-- SessionType (Citrix.Broker.Admin.SDK.SessionType?)

Session property indicating the type of the current session.

Session properties are always null for multi-session machines.

-- SessionUid (System.Int64?)

Session property indicating the UID of the current session.

Session properties are always null for multi-session machines.

-- SessionUserName (System.String)

Session property indicates the name of the current session's user (in the form DOMAIN\user).

Session properties are always null for multi-session machines.

-- SessionUserSID (System.String)

Session property indicates the SID of the current session's user.

Session properties are always null for multi-session machines.

-- SID (System.String)

The SID of the machine.

-- SummaryState (Citrix.Broker.Admin.SDK.DesktopSummaryState)

Indicates the overall state of the desktop associated with the machine. The overall state is a result of other more specific states such as session state, registration state and power state. Possible values: Off, Unregistered, Available, Disconnected, InUse, Preparing.

-- SupportedPowerActions (System.String[])

A list of power actions supported by this machine.

-- Tags (System.String[])

A list of tags for the machine.

-- Uid (System.Int32)

UID of the machine object.

-- UUID (System.Guid)

UUID of the machine object.

-- VMToolsState (Citrix.Broker.Admin.SDK.VMToolsState)

State of the hypervisor tools present on the VM (if any).

Possible values are:

NotPresent, Unknown, NotStarted, Running.

-- WillShutdownAfterUse (System.Boolean)

Flag indicating if this machine is tainted and will be shut down after all sessions on the machine have ended. This flag is only ever true on power-managed, single-session machines.

Note: The machine will not shut down if it is in maintenance mode; it will shut down only after it is taken out of maintenance mode.

-- WindowsConnectionSetting (Citrix.Broker.Admin.SDK.WindowsConnectionSetting?)

The logon mode reported by Windows itself (multi-session machines only). For single-session machines the value is always

hardwired to LogonEnabled.

Possible values are:

LogonEnabled, Draining, DrainingUntilRestart and LogonDisabled.

Related topics

[Group-BrokerMachine](#)

Parameters

-Uid<Int32>

Gets a machine with a specific UID.

Required?	true
Default Value	
Accept Pipeline Input?	false

-MachineName<String>

Gets machines with a specific machine name (in the form domain\machine).

Required?	false
Default Value	
Accept Pipeline Input?	false

-AgentVersion<String>

Gets machines with a specific Citrix Virtual Delivery Agent version.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AllocationType<AllocationType>

Gets machines from catalogs with the specified allocation type.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ApplicationInUse<String>

Gets machines running a specified published application (identified by browser name).

String comparisons are case-insensitive.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssignedClientName<String>

Gets machines that have been assigned to the specific client name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssignedIPAddress<String>

Gets machines that have been assigned to the specific IP address.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedUserFullName<String>

Gets machines with an associated user identified by their full name (usually 'first-name last-name').

Associated users are all current users of a desktop, plus the assigned users for private desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedUserName<String>

Gets machines with an associated user identified by their user name (in the form 'domain\user').

Associated users are all current users of a desktop, plus the assigned users for private desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedUserSID<String>

Gets machines with an associated user identified by their Windows SID.

Associated users are all current users of a desktop, plus the assigned users for private desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedUserUPN<String>

Gets machines with an associated user identified by their User Principle Name (in the form 'user@domain').

Associated users are all current users of a desktop, plus the assigned users for private desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-BrowserName<String>

Gets assigned machines backing desktop resources that have browser names matching the specified name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-CatalogName<String>

Gets machines from the catalog with the specific name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-CatalogUid<Int32>

Gets machines from the catalog with the specific UID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-CatalogUUID<Guid>

Gets machines from the catalog with the specific UUID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ColorDepth<ColorDepth>

Gets machines configured with a specific color depth.

Valid values are FourBit, EightBit, SixteenBit, and TwentyFourBit.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ControllerDNSName<String>

Gets machines with a specific DNS name of the controller they are registered with.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DeliveryType<DeliveryType>

Gets machines of a particular delivery type.

Valid values are AppsOnly, DesktopsOnly, DesktopsAndApps

Required?	false
Default Value	
Accept Pipeline Input?	false

-Description<String>

Gets machines with a specific description.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopCondition<String>

Gets machines with an outstanding desktop condition.

Valid values are:

- o CPU: Indicates the machine has high CPU usage
- o ICALatency: Indicates the network latency is high
- o UPMLogonTime: Indicates that the profile load time was high

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopGroupName<String>

Gets machines from a desktop group with the specified name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopGroupUid<Int32>

Gets machines from a desktop group with a specific UID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopGroupUUID<Guid>

Gets machines from a desktop group with a specific UUID.

Required?	false

Default Value	
Accept Pipeline Input?	false

-DesktopKind<DesktopKind>

Deprecated: Use AllocationType parameter.

Gets machines of a particular kind.

Valid values are Private, Shared.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopUid<Int32>

Gets the machine that corresponds to the desktop with the specific UID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DNSName<String>

Gets machines with the specific DNS name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-FaultState<MachineFaultState>

Gets machines currently in the specified fault state.

Required?	false
Default Value	
Accept Pipeline Input?	false

-FunctionalLevel<FunctionalLevel>

Gets machines with a specific FunctionalLevel.

Valid values are L5, L7, L7_6

Required?	false
Default Value	
Accept Pipeline Input?	false

-HostedMachineId<String>

Gets machines with the specific machine ID known to the hypervisor.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HostedMachineName<String>

Gets machines with the specific machine name known to the hypervisor.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HostingServerName<String>

Gets machines by the name of the hosting hypervisor server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HypervisorConnectionName<String>

Gets machines with the specific name of the hypervisor connection hosting them.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HypervisorConnectionUid<Int32>

Gets machines with the specific UID of the hypervisor connection hosting them.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HypHypervisorConnectionUid<Guid>

Gets machines with the specific UUID of the hypervisor connection hosting them.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IconUid<Int32>

Gets machines by configured icon. Note that machines with a null IconUid use the icon of the desktop group.

Required?	false

Default Value	
Accept Pipeline Input?	false

-ImageOutOfDate<Boolean>

Gets machines depending on whether their disk image is out of date or not (for machines provisioned using MCS only).

Required?	false
Default Value	
Accept Pipeline Input?	false

-InMaintenanceMode<Boolean>

Gets machines by whether they are in maintenance mode or not.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IPAddress<String>

Gets machines with a specific IP address.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IsAssigned<Boolean>

Gets machines according to whether they are assigned or not. Machines may be assigned to one or more users or groups, a client IP address or a client endpoint name.

Required?	false

Default Value	
Accept Pipeline Input?	false

-IsPhysical<Boolean>

Gets machines according to whether they can be power managed by XenDesktop or not.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastConnectionFailure<ConnectionFailureReason>

Gets machines with a specific reason for the last recorded connection failure. This value is None if the last connection was successful or if there has been no attempt to connect to the desktop yet.

Valid values are None, SessionPreparation, RegistrationTimeout, ConnectionTimeout, Licensing, Ticketing, and Other.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastConnectionTime<DateTime>

Gets machines on which a user session connection occurred at a specific time. This is the time at which the broker detected that the connection attempt either succeeded or failed.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastConnectionUser<String>

Gets machines where a specific user name last attempted a connection (in the form 'domain\user').

Required?	false

Default Value	
Accept Pipeline Input?	false

-LastDeregistrationReason<DeregistrationReason>

Gets machines whose broker last recorded a specific deregistration reason.

Valid values are \$null, AgentShutdown, AgentSuspended, AgentRequested, IncompatibleVersion, AgentAddressResolutionFailed, AgentNotContactable, AgentWrongActiveDirectoryOU, EmptyRegistrationRequest, MissingRegistrationCapabilities, MissingAgentVersion, InconsistentRegistrationCapabilities, NotLicensedForFeature, UnsupportedCredentialSecurityVersion, InvalidRegistrationRequest, SingleMultiSessionMismatch, FunctionalLevelTooLowForCatalog, FunctionalLevelTooLowForDesktopGroup, PowerOff, DesktopRestart, DesktopRemoved, AgentRejectedSettingsUpdate, SendSettingsFailure, SessionAuditFailure, SessionPrepareFailure, ContactLost, SettingsCreationFailure, UnknownError and BrokerRegistrationLimitReached.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastDeregistrationTime<DateTime>

Gets machines by the time that they were last deregistered.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastErrorReason<String>

Gets machines with the specified last error reason.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastErrorTime<DateTime>

Gets machines with the specified last error time.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastHostingUpdateTime<DateTime>

Gets machines with a specific time that the hosting information was last updated.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastPvdErrorReason<String>

Gets machines with the specified last Personal vDisk preparation error reason.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastPvdErrorTime<DateTime>

Gets machines with the specified last Personal vDisk preparation error time.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LoadIndex<Int32>

Gets machines by their current load index.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MachineInternalState<MachineInternalState>

Gets machines with the specified internal state.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Metadata<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

Required?	false
Default Value	
Accept Pipeline Input?	false

-OSType<String>

Gets machines by the type of operating system they are running.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OSVersion<String>

Gets machines by the version of the operating system they are running.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PersistUserChanges<PersistUserChanges>

Gets machines by the location where the user changes are persisted.

- o OnLocal - User changes are persisted locally.
- o Discard - User changes are discarded.
- o OnPvd - User changes are persisted on the Pvd.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PowerActionPending<Boolean>

Gets machines depending on whether a power action is pending or not.

Valid values are \$true or \$false.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PowerState<PowerState>

Gets machines with a specific power state.

Valid values are Unmanaged, Unknown, Unavailable, Off, On, Suspended, TurningOn, TurningOff, Suspending, and Resuming.

Required?	false

Default Value	
Accept Pipeline Input?	false

-ProvisioningType<ProvisioningType>

Gets machines that are in a catalog with a particular provisioning type. Values can be:

- o Manual - No provisioning.
- o PVS - Machine provisioned by PVS (machine may be physical, blade, VM,...).
- o MCS - Machine provisioned by MCS (machine must be VM).

Required?	false
Default Value	
Accept Pipeline Input?	false

-PublishedApplication<String>

Gets machines with a specific application published to them (identified by its browser name).

Required?	false
Default Value	
Accept Pipeline Input?	false

-PublishedName<String>

Gets desktops with a specific published name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PvdEstimatedCompletionTime<DateTime>

If preparation of the Personal vDisk is currently in progress for this machine, this reports an estimation of the time at which the process will be complete.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PvdPercentDone<Int32>

Gets machines a specific percentage through the Personal vDisk preparation process.

This property is typically used with advanced filtering; see about_Broker_Filtering.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PvdStage<PvdStage>

Gets machines at a specific personal vDisk stage.

Valid values are None, Requested, Starting, Working and Failed.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PvdUpdateStartTime<DateTime>

If preparation of the Personal vDisk is currently in progress for this machine, this reports when the update process began.

Required?	false
Default Value	
Accept Pipeline Input?	false

-RegistrationState<RegistrationState>

Gets machines in a specific registration state.

Valid values are Unregistered, Initializing, Registered, and AgentError.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ScheduledReboot<ScheduledReboot>

Gets machines according to their current status with respect to any scheduled reboots (for either scheduled desktop group reboots or image rollout purposes). Valid values are:

- o None - No reboot currently scheduled.
- o Pending - Reboot scheduled but machine still available for use.
- o Draining - Reboot scheduled. New logons are disabled, but reconnections to existing sessions are allowed.
- o InProgress - Machine is actively being rebooted.
- o Natural - Natural reboot in progress. Machine is awaiting a restart.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SecureIcaRequired<Boolean>

Gets machines configured with a particular SecureIcaRequired setting. Note that the machine setting of \$null indicates that the desktop group value is used.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionAutonomouslyBrokered<Boolean>

Gets machines according to whether their current session is autonomously brokered or not. Autonomously brokered sessions are HDX sessions established by direct connection without being brokered.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionClientAddress<String>

Gets machines with a specific client IP address.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionClientName<String>

Gets machines with a specific client name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionClientVersion<String>

Gets machines with a specific client version.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionConnectedViaHostName<String>

Gets machines with a specific incoming connection host name. This is usually a proxy or Citrix Access Gateway server.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionConnectedViaIP<String>

Gets machines with a specific incoming connection IP address.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionCount<Int32>

Gets machines according to the total number of both pending and established user sessions on the machine.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionDeviceId<String>

Gets machines with a specific client device ID.

Required?	false
Default Value	
Accept Pipeline Input?	

Accept Pipeline Input?	false

-SessionHardwareId<String>

Gets machines with a specific client hardware ID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionHidden<Boolean>

Gets machines depending on whether their sessions are hidden or not. Hidden sessions are treated as though they do not exist when launching sessions using XenDesktop; a hidden session cannot be reconnected to, but a new session may be launched using the same entitlement.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionKey<Guid>

Gets machine running the session with a specified unique key.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionLaunchedViaHostName<String>

Gets machines with a specific host name of the StoreFront server from which the user launched the session.

Session properties are always null for multi-session machines.

Required?	false

Default Value	
Accept Pipeline Input?	false

-SessionLaunchedViaIP<String>

Gets machines with a specific IP address of the StoreFront server from which the user launched the session.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionProtocol<String>

Gets machines with connections using a specific protocol, for example HDX, RDP, or Console.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionSecureIcaActive<Boolean>

Gets machines depending on whether the current session uses SecureICA or not.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionsEstablished<Int32>

Gets machines according to the number of established user sessions present on the machine.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionSmartAccessTag<String>

Gets machines where the session has the specific SmartAccess tag.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionsPending<Int32>

Get machines according to the number of pending user sessions for the machine.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionStartTime<DateTime>

Gets machines with a specific session start time.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionState<SessionState>

Gets machines with a specific session state.

Valid values are \$null, Other, PreparingSession, Connected, Active, Disconnected, Reconnecting, NonBrokeredSession, and Unknown.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionStateChangeTime<DateTime>

Gets machines whose sessions last changed state at a specific time.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionSupport<SessionSupport>

Gets machines that have the specified session capability. Values can be:

- o SingleSession - Single-session only machine.
- o MultiSession - Multi-session capable machine.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionType<SessionType>

Gets machines with a specific session state.

Session properties are always null for multi-session machines.

Required?	false

Default Value	
Accept Pipeline Input?	false

-SessionUid<Int64>

Gets machines with a specific session UID (\$null for no session).

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionUserName<String>

Gets machines with a specific user name for the current session (in the form 'domain\user').

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionUserSID<String>

Gets machines with a specific SID of the current session user.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SID<String>

Gets machines with a specific machine SID.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SummaryState<DesktopSummaryState>

Gets machines with a specific summary state.

Valid values are Off, Unregistered, Available, Disconnected, and InUse.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SupportedPowerActions<String[]>

A list of power actions supported by this machine.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Tag<String>

Gets machines where the session has the given SmartAccess tag.

Required?	false
Default Value	
Accept Pipeline Input?	false

-UUID<Guid>

Gets machines with the specified value of UUID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-VMToolsState<VMToolsState>

Gets machines with a specific VM tools state.

Valid values are NotPresent, Unknown, NotStarted, and Running.

Required?	false
Default Value	
Accept Pipeline Input?	false

-WillShutDownAfterUse<Boolean>

Gets machines depending on whether they shut down after use or not.

Required?	false
Default Value	
Accept Pipeline Input?	false

-WindowsConnectionSetting<WindowsConnectionSetting>

Gets machines according to their current Windows connection setting (logon mode). Valid values are:

- o LogonEnabled - All logons are enabled.
- o Draining - New logons are disabled, but reconnections to existing sessions are allowed.
- o DrainingUntilRestart - Same as Draining, but setting reverts to LogonEnabled when machine next restarts.
- o LogonDisabled - All logons and reconnections are disabled.

This is a Windows setting and is not controlled by XenDesktop. It applies only to multi-session machines; for single-session

machines its value is always LogonEnabled.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssignedUserSID<String>

Gets machines with the specific SID of the user to whom the desktop is assigned.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about_Broker_Filtering for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by - ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell style filter expression. See about_Broker_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.Machine

Get-BrokerMachine returns an object for each matching desktop.

Notes

It is generally better to compare dates and times using -Filter and relative comparisons. See about_Broker_Filtering and the examples in this topic for more information.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Get-BrokerMachine -PowerState Suspended
```

```
C:\PS> Get-BrokerMachine -Filter { PowerState -eq 'Suspended' }
```

These commands return all suspended machines. The second form uses advanced filtering (see about_Broker_Filtering).

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerMachine -DNSName '*.mydomain.mycompany.com'
```

This command returns all machines belonging to the DNS domain mydomain.mycompany.com.

----- EXAMPLE 3 -----

```
C:\PS> Get-BrokerMachine -Filter { RegistrationState -eq 'Registered' -and HypervisorConnectionUid -eq 5 }
```

This command returns all registered machines running on the specified hypervisor connection.

----- EXAMPLE 4 -----

```
C:\PS> Get-BrokerMachine -MachineName 'MyDomain\X*' | Remove-BrokerDesktopGroup -DesktopGroup 2
```

This command finds all of the machines in MyDomain with names beginning with X and removes them from the specified desktop group.

----- EXAMPLE 5 -----

```
C:\PS> Get-BrokerMachine -Filter { DesktopGroupUid -ne $null }
```

This command gets all desktops in a site. Use this instead of the deprecated Get-BrokerDesktop command.

Get-BrokerMachineCommand

May 04, 2016

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

May 04, 2016

Gets machine configurations defined for this site.

Syntax

```
Get-BrokerMachineConfiguration [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerMachineConfiguration [[-Name] <String>] [-ConfigurationSlotUid <Int32>] [-LeafName <String>] [-Metadata <String>] [-DesktopGroupUid <Int32>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Retrieves machine configurations matching the specified criteria. If no parameters are specified this cmdlet enumerates all machine configurations.

Machine configurations contain binary arrays of settings data that are managed using SDK snap-ins. Each machine configuration is associated with a configuration slot and referenced by Name. The configuration slot restricts the settings that can be held by the machine configuration. For example, only configurations for Citrix User Profile Manager can be associated with the "User Profile Manager" slot.

See `about_Broker_Filtering` for information about advanced filtering options.

----- BrokerMachineConfiguration Object

The machine configuration object returned represents a named collection of related settings values that are applied to a desktop group.

-- ConfigurationSlotUid (System.Int32)

Uid of the associated configuration slot.

-- Description (System.String)

Optional description of the machine configuration.

-- DesktopGroupUids (System.Int32[])

List of desktop group Uids that this machine configuration has been added to.

-- LeafName (System.String)

Name of this machine configuration.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Map of metadata associated with this machine configuration.

-- Name (System.String)

Unique "SlotName\MachineConfigurationName" for this machine configuration.

-- Policy (System.Byte[])

A binary array of encoded settings.

-- Uid (System.Int32)

Uid of this machine configuration.

Related topics

[New-BrokerMachineConfiguration](#)

[Set-BrokerMachineConfiguration](#)

[Rename-BrokerMachineConfiguration](#)

[Remove-BrokerMachineConfiguration](#)

[Add-BrokerMachineConfiguration](#)

Parameters

-Uid<Int32>

Get only the machine configuration with the specified unique identifier.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Name<String>

Get only the machine configuration with the specified name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ConfigurationSlotUid<Int32>

Get only the machine configurations associated with the specified configuration slot.

Required?	false

Default Value	
Accept Pipeline Input?	false

-LeafName<String>

Get only the machine configurations that have the specified leaf name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Metadata<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopGroupUid<Int32>

Get only the machine configurations that have been assigned to the specified desktop group.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about_Broker_Filtering for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by - ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell style filter expression. See `about_Broker_Filtering` for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through `Select-Object`, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

`Citrix.Broker.Admin.SDK.MachineConfiguration`

`Get-BrokerMachineConfiguration` returns an object for each matching machine configuration.

Examples

----- **EXAMPLE 1** -----

`Get-BrokerMachineConfiguration`

Retrieves a list of every defined machine configuration.

----- **EXAMPLE 2** -----

Get-BrokerMachineConfiguration -Name Receiver\Engineering

Retrieves the machine configuration named "Receiver\Engineering".

----- **EXAMPLE 3** -----

Get-BrokerMachineConfiguration -Name UPM*

Retrieves a list of every machine configuration associated with the configuration slot named "UPM".

----- **EXAMPLE 4** -----

Get-BrokerMachineConfiguration -LeafName "Dept*"

Retrieves a list of every machine configuration with a LeafName that starts with "Dept", regardless of the associated configuration slot.

Get-BrokerMachineStartMenuShortcutIcon

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

UnknownObject

One of the specified objects was not found.

PartialData

Only a subset of the available data was returned.

InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

CouldNotQueryDatabase

The query required to get the database was not defined.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-BrokerScopedObject -ObjectType Scheme
```

```
ScopeId   : eff6f464-f1ee-4442-add3-99982e0cec01
ScopeName  : Sales
ObjectType : Scheme
ObjectId   : cd4174ee-9e4b-4e57-b126-9dbf757fe493
ObjectName : MyExampleScheme
Description : Test scheme
```

ScopeId : 304e0fa7-d390-47f0-a94f-7e956a324c41
ScopeName : Finance
ObjectType : Scheme
ObjectId : cd4174ee-9e4b-4e57-b126-9dbf757fe493
ObjectName : MyExampleScheme
Description : Test scheme

ScopeId :
ScopeName :
ObjectType : Scheme
ObjectId : 5062e46b-71bc-4ac9-901a-30fe6797e2f6
ObjectName : AnotherScheme
Description : Another scheme in no scopes

Gets all of the scoped objects with type Scheme. The example output shows a scheme object (MyExampleScheme) in two scopes Sales and Finance, and another scheme (AnotherScheme) that is not in any scope. The ScopeId and ScopeName values returned are null in the final record.

Get-BrokerServiceAddedCapability

May 04, 2016

Gets any added capabilities for the Broker Service on the controller.

Syntax

```
Get-BrokerServiceAddedCapability [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Enables updates to the Broker Service on the controller to be detected.

There is no requirement for a database connection to be configured in order for this command to be used.

Related topics

Parameters

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

System.String

String containing added capabilities.

Notes

If the command fails, the following errors can be returned.

Error Codes

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
C:\PS>Get-BrokerServiceAddedCapability
```

Get the added capabilities of the Broker Service.

Get-BrokerServiceInstance

May 04, 2016

Gets the service instance entries for the Broker Service.

Syntax

```
Get-BrokerServiceInstance [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Returns service interfaces published by instances of the Broker Service. Each instance of a service publishes multiple interfaces with distinct interface types, and each of these interfaces is represented as a ServiceInstance object. Service instances can be used to register the service with a central configuration service so that other services can use the functionality.

You do not need to configure a database connection to use this command.

Related topics

[Get-BrokerServiceStatus](#)

[Reset-BrokerServiceGroupMembership](#)

Parameters

-AdminAddress <String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Return Values

Citrix.Broker.Sdk.ServiceInstance

The Get-BrokerServiceInstance command returns an object containing the following properties.

ServiceGroupUid <Guid>

Specifies the unique identifier for the service group of which the service is a member.

ServiceGroupName <String>

Specifies the name of the service group of which the service is a member.

ServiceInstanceUID <Guid>

Specifies the unique identifier for registered service instances, which are service instances held by and obtained from a central configuration service. Unregistered service instances do not have unique identifiers.

ServiceType <String>

Specifies the service instance type. For this service, the service instance type is always Broker.

Address

Specifies the address of the service instance. The address can be used to access the service and, when registered in the central configuration service, can be used by other services to access the service.

Binding

Specifies the binding type that must be used to communicate with the service instance. In this release of XenDesktop, the binding type is always 'wcf_HTTP_kerb'. This indicates that the service provides a Windows Communication Foundation endpoint that uses HTTP binding with integrated authentication.

Version

Specifies the version of the service instance. The version number is used to ensure that the correct versions of the services are used for communications.

ServiceAccount <String>

Specifies the Active Directory account name for the machine on which the service instance is running. The account name is used to provide information about the permissions required for interservice communications.

ServiceAccountSid <String>

Specifies the Active Directory account security identifier for the machine on which the service instance is running.

InterfaceType <String>

Specifies the interface type. Each service can provide multiple service instances, each for a different purpose, and the interface defines the purpose. Available interfaces are:

SDK - for PowerShell operations

InterService - for operations between different services

Peer - for communications between services of the same type

Metadata <Citrix.Broker.Sdk.Metadata[]>

The collection of metadata associated with registered service instances, which are service instances held by and obtained from a central configuration service. Metadata is not stored for unregistered service instances.

Notes

If the command fails, the following errors can be returned.

Error Codes

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

c:\PS>Get-BrokerServiceInstance

Get all instances of the Broker Service running on the specified machine. For remote services, use the AdminAddress parameter to define the service for which the interfaces are required. If the AdminAddress parameter has not been specified for the runspace, service instances running on the local machine are returned.

Get-BrokerServiceStatus

May 04, 2016

Gets the current state of the Broker Service on the controller.

Syntax

```
Get-BrokerServiceStatus [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Enables the status of the Broker Service on the controller to be determined. If the service has multiple data stores it will return the overall state as an aggregate of all the data store states. For example, if the site data store status is OK and the secondary data store status is DBUnconfigured then it will return DBUnconfigured. Before using this command, you don't have to configure the database connection to the Service.

Related topics

[Set-BrokerDBConnection](#)

[Test-BrokerDBConnection](#)

[Get-BrokerDBConnection](#)

[Get-BrokerDBSchema](#)

Parameters

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Get-BrokerServiceStatus command returns an object containing the status of the Broker Service together with extra diagnostics information.

DBUnconfigured

The Broker Service does not have a database connection configured.

DBRejectedConnection

The database rejected the logon attempt from the Broker Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the Broker Service schema has not been added to the database.

DBNotFound

The specified database could not be located with the configured connection string.

DBMissingOptionalFeature

The Broker is connected to a database that is valid, but it does not have the full functionality required for optimal performance. Upgrading the database is advisable.

DBMissingMandatoryFeature

The Broker is connected to a database that is valid, but it does not have the full functionality required so the Broker cannot function. Upgrading the database is required.

DBNewerVersionThanService

The version of the Broker Service currently in use is incompatible with the version of the Broker Service schema on the database. Upgrade the Broker Service to a more recent version.

DBOlderVersionThanService

The version of the Broker Service schema on the database is incompatible with the version of the Broker Service currently in use. Upgrade the database schema to a more recent version.

DBVersionChangeInProgress

A database schema upgrade is currently in progress.

OK

The Broker Service is running and is connected to a database containing a valid schema.

PendingFailure

Connectivity between the Broker Service and the database has been lost. This may be a transitory network error, but may indicate a loss of connectivity that requires administrator intervention.

Failed

Connectivity between the Broker and the database has been lost for an extended period of time, or has failed due to a configuration problem. The Broker service cannot operate while its connection to the database is unavailable.

Unknown

The service status cannot be determined.

Notes

If the command fails, the following errors can be returned.

Error Codes

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

C:\PS>Get-BrokerServiceStatus

Get the current status of the Broker Service.

Get-BrokerSession

May 04, 2016

Gets a list of sessions.

Syntax

```
Get-BrokerSession [-Uid] <Int64> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerSession [[-SessionKey] <Guid>] [-AgentVersion <String>] [-ApplicationInUse <String>] [-AppState  
<SessionAppState>] [-AppStateLastChangeTime <DateTime>] [-AutonomouslyBrokered <Boolean>] [-BrokeringDuration  
<Int32>] [-BrokeringTime <DateTime>] [-BrokeringUserName <String>] [-BrokeringUserSID <String>] [-CatalogName  
<String>] [-ClientAddress <String>] [-ClientName <String>] [-ClientPlatform <String>] [-ClientProductId <Int32>] [-ClientVersion  
<String>] [-ConnectedViaHostName <String>] [-ConnectedViaIP <String>] [-ConnectionMode <ConnectionMode>] [-  
ControllerDNSName <String>] [-DesktopGroupName <String>] [-DesktopGroupUid <Int32>] [-DesktopKind <DesktopKind>] [-  
DesktopSID <String>] [-DesktopUid <Int32>] [-DeviceId <String>] [-DNSName <String>] [-EstablishmentDuration <Int32>] [-  
EstablishmentTime <DateTime>] [-HardwareId <String>] [-Hidden <Boolean>] [-HostedMachineName <String>] [-  
HostingServerName <String>] [-HypervisorConnectionName <String>] [-ImageOutOfDate <Boolean>] [-InMaintenanceMode  
<Boolean>] [-IPAddress <String>] [-IsAnonymousUser <Boolean>] [-IsPhysical <Boolean>] [-LaunchedViaHostName  
<String>] [-LaunchedViaIP <String>] [-LogoffInProgress <Boolean>] [-LogonInProgress <Boolean>] [-MachineName <String>]  
[-MachineSummaryState <DesktopSummaryState>] [-MachineUid <Int32>] [-Metadata <String>] [-OSType <String>] [-  
PersistUserChanges <PersistUserChanges>] [-PowerState <PowerState>] [-Protocol <String>] [-ProvisioningType  
<ProvisioningType>] [-ReceiverIPAddress <String>] [-ReceiverName <String>] [-SecureIcaActive <Boolean>] [-SessionId  
<Int32>] [-SessionState <SessionState>] [-SessionStateChangeTime <DateTime>] [-SessionSupport <SessionSupport>] [-  
SessionType <SessionType>] [-StartTime <DateTime>] [-UntrustedUserName <String>] [-UserFullName <String>] [-  
UserName <String>] [-UserSID <String>] [-UserUPN <String>] [-ApplicationUid <Int32>] [-SharedDesktopUid <Int32>] [-  
ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>]  
[-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Retrieves sessions matching all the specified criteria. If no parameters are specified this cmdlet enumerates all sessions.

----- BrokerSession Object

The session object returned represents a session on a machine in the site. The session could be for a desktop or application

-- AgentVersion (System.String)

Version of the Citrix Virtual Delivery Agent (VDA) installed on the machine.

-- ApplicationsInUse (System.String[])

List of applications in use in the session. Applications are identified by their administrative name.

-- AppState (Citrix.Broker.Admin.SDK.SessionAppState)

The app state of the session. Valid values are PreLogon, PreLaunched, Active, Desktop, Lingering and NoApps.

-- AppStateLastChangeTime (System.DateTime?)

The time when the session entered the current app state.

-- AutonomouslyBrokered (System.Boolean)

Indicates whether this is an HDX session established by direct connection without being brokered.

-- BrokeringDuration (System.Int32?)

Time taken to broker the session (in milliseconds).

-- BrokeringTime (System.DateTime?)

Time at which the session was brokered.

-- BrokeringUserName (System.String)

The user name of the brokering user.

-- BrokeringUserSID (System.String)

The SID of the brokering user.

-- CatalogName (System.String)

The name of the catalog that the machine hosting the session is assigned to.

-- ClientAddress (System.String)

The IP address of the client connected to the session.

-- ClientName (System.String)

The host name of the client connected to the session.

-- ClientPlatform (System.String)

The name of client platform, as indicated by client product ID.

-- ClientProductId (System.Int32?)

The product ID of the client connected to the session.

-- ClientVersion (System.String)

The version of the Citrix Receiver running on the client connected to the session.

-- ConnectedViaHostName (System.String)

The host name of the incoming connection. This is usually a gateway, router or client.

-- ConnectedViaIP (System.String)

The IP address of the incoming connection This is usually a gateway, router or client.

-- ConnectionMode (Citrix.Broker.Admin.SDK.ConnectionMode?)

The way in which the most recent connection to the session was established. Valid modes are:

- o Brokered: established through the XenDesktop Broker (for example, using StoreFront).
- o Unbrokered: direct connection (for example, using the console or direct RDP/HDX).
- o LeasedConnection: established through the XenDesktop Broker using a connection lease.
- o VdaHighAvailabilityMode: direct connection while VDA in high-availability mode.
- o ThirdPartyBroker: established through a third-party Broker.
- o ThirdPartyBrokerWithLicensing: established and licensed through a third-party Broker.

-- ControllerDNSName (System.String)

The DNS host name of the controller that the session's hosting machine is registered with.

-- DesktopGroupName (System.String)

Name of the desktop group of the machine the session is on.

-- DesktopGroupUid (System.Int32)

UID of the desktop group of the machine the session is on.

-- DesktopKind (Citrix.Broker.Admin.SDK.DesktopKind)

Indicates if the session is shared or private.

-- DesktopSID (System.String)

The Windows SID of the machine the session is on.

-- DesktopUid (System.Int32)

For a desktop session, the unique identifier of the desktop.

-- DeviceId (System.String)

Unique identifier for the client device that has most recently been associated with the session.

-- DNSName (System.String)

The DNS host name of the machine hosting the session.

-- EstablishmentDuration (System.Int32?)

Duration that it took to establish the session.

-- EstablishmentTime (System.DateTime?)

Time at which the session was established.

-- HardwareId (System.String)

Unique identifier for the client hardware that has been most recently associated with the session.

-- Hidden (System.Boolean)

Flag to indicate if the session is currently hidden from the user and not to be reconnected to.

-- HostedMachineName (System.String)

The friendly name of a hosted machine running the session, as used by its hypervisor. This does not necessarily match either the DNS or AD name of the machine.

-- HostingServerName (System.String)

DNS name of the hypervisor that is hosting the machine hosting the session.

-- HypervisorConnectionName (System.String)

The name of the hypervisor connection that the machine hosting the session has been assigned to.

-- ImageOutOfDate (System.Boolean?)

Denotes whether the VM image for a hosted machine is out of date and due to be updated to a new master image when the machine next reboots.

-- InMaintenanceMode (System.Boolean)

Denotes whether the machine hosting the session is in maintenance mode.

-- IPAddress (System.String)

The IP address of the machine hosting the session.

-- IsAnonymousUser (System.Boolean)

Indicates whether the session was established anonymously (without user credentials), in this case a temporary local user account on the machine is used.

-- IsPhysical (System.Boolean)

This value is false if the machine hosting the session can be power managed, and true otherwise

-- LaunchedViaHostName (System.String)

The host name of the StoreFront server used to launch the session.

-- LaunchedViaIP (System.String)

The IP address of the StoreFront server used to launch the session.

-- LogoffInProgress (System.Boolean)

Indicates whether the session is in the process of being logged off.

-- LogonInProgress (System.Boolean)

Indicates whether the session is still executing user logon processing or not.

-- MachineName (System.String)

DNS host name of the machine hosting the session.

-- MachineSummaryState (Citrix.Broker.Admin.SDK.DesktopSummaryState)

The summary state of the machine (will be Unregistered, Disconnected, or InUse)

-- MachineUid (System.Int32)

UID of the machine hosting the session.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Map of metadata for this session.

-- OSType (System.String)

A string that can be used to identify the operating system that is running on the machine hosting the session.

-- PersistUserChanges (Citrix.Broker.Admin.SDK.PersistUserChanges)

Describes whether/how the user changes are persisted. Possible values are:

- o OnLocal - Persist the user changes locally.

- o Discard - Discard user changes.

- o OnPvd - Persist user changes on the Citrix Personal vDisk.

-- PowerState (Citrix.Broker.Admin.SDK.PowerState)

The current power state of the machine hosting the session. Possible values are: Unmanaged, Unknown, Unavailable, On, Suspended, TurningOn, TurningOff, Suspending and Resuming.

-- Protocol (System.String)

The protocol that the session is using, can be HDX, RDP, or Console. Console sessions on XenDesktop 5 VDAs appear with a blank protocol.

-- ProvisioningType (Citrix.Broker.Admin.SDK.ProvisioningType)

Describes how the machine hosting the session was provisioned, possible values are:

- o Manual: No provisioning.

- o PVS: Machine provisioned by PVS (may be physical, blade, VM,...)

- o MCS: Machine provisioned by MCS (machine must be VM)

-- ReceiverIPAddress (System.String)

The IP address of the client as supplied by Receiver (for example, StoreFront) when the session was launched, or reconnected.

-- ReceiverName (System.String)

The name of the client as supplied by Receiver (for example, StoreFront) when the session was launched, or reconnected.

-- SecureIcaActive (System.Boolean?)

Indicates whether SecureICA is active on the session.

-- SessionId (System.Int32)

Deprecated. A unique identifier that Remote Desktop Services uses to track the session but it is only unique on that machine.

-- SessionKey (System.Guid)

GUID that provides a unique identifier for this session.

-- SessionState (Citrix.Broker.Admin.SDK.SessionState)

The state of the session. Valid values are Connected, Active or Disconnected.

For a session on a machine with functional level below L7, the additional states PreparingSession, Reconnecting, NonBrokeredSession, Other, and Unknown can also occur.

-- SessionStateChangeTime (System.DateTime)

The time of the most recent state change for the session.

-- SessionSupport (Citrix.Broker.Admin.SDK.SessionSupport)

Indicates if the machine hosting the session supports multiple or single sessions.

-- SessionType (Citrix.Broker.Admin.SDK.SessionType)

Indicates if this is an Application or Desktop session.

-- SmartAccessTags (System.String[])

The Smart Access tags for this session.

-- StartTime (System.DateTime?)

Indicates when the session was started.

-- Uid (System.Int64)

Unique identifier of this session.

-- UntrustedUserName (System.String)

The name of the logged-on user reported directly from the machine (in the form DOMAIN\user). This may be useful where the user is logged in to a non-domain account, however the name cannot be verified and must therefore be considered untrusted.

-- UserFullName (System.String)

The full name of the user.

-- UserName (System.String)

The name of the user.

-- UserSID (System.String)

The user's Windows SID.

-- UserUPN (System.String)

The user's User Principal Name

Related topics

[Disconnect-BrokerSession](#)

[Stop-BrokerSession](#)

[Get-BrokerDesktop](#)

Parameters

-Uid<Int64>

Get session by its Uid.

Required?	true
Default Value	
Accept Pipeline Input?	false

-SessionKey<Guid>

Gets session having the specified unique key.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AgentVersion<String>

Gets sessions with a specific Virtual Desktop Agent version.

Required?	false

Default Value	
Accept Pipeline Input?	false

-ApplicationInUse<String>

Gets sessions running specific applications (identified by their SDK Name property).

Required?	false
Default Value	
Accept Pipeline Input?	false

-AppState<SessionAppState>

Get sessions by their app state.

Valid values are PreLogon, PreLaunched, Active, Desktop, Lingering and NoApps.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AppStateLastChangeTime<DateTime>

Get sessions by their app state change time.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AutonomouslyBrokered<Boolean>

Gets sessions according to whether they are autonomously brokered or not. Autonomously brokered sessions are HDX sessions established by direct connection without being brokered.

Required?	false

Default Value	
Accept Pipeline Input?	false

-BrokeringDuration<Int32>

Gets session with a specific time taken to broker. In general, Citrix recommends using -Filter and relative comparisons.

Required?	false
Default Value	
Accept Pipeline Input?	false

-BrokeringTime<DateTime>

Get sessions brokered at a specific time. In general, Citrix recommends using -Filter and relative comparisons.

Required?	false
Default Value	
Accept Pipeline Input?	false

-BrokeringUserName<String>

Get sessions by brokering user.

Required?	false
Default Value	
Accept Pipeline Input?	false

-BrokeringUserSID<String>

Get sessions by brokering user SID.

Required?	false
Default Value	

Accept Pipeline Input?	false

-CatalogName<String>

Gets sessions on machines from a specific catalog name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ClientAddress<String>

Get sessions by client IP address.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Client Name<String>

Get sessions by client name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ClientPlatform<String>

Get sessions by client platform.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ClientProductId<Int32>

Get sessions by client product ID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ClientVersion<String>

Get sessions by client version.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ConnectedViaHostName<String>

Get sessions by host name of the incoming connection. This is usually a proxy or Citrix Access Gateway server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ConnectedViaIP<String>

Get sessions by IP address of the incoming connection.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ConnectionMode<ConnectionMode>

Gets sessions by the way in which the most recent connection to the session was established.

Valid modes are Brokered, Unbrokered, LeasedConnection, VdaHighAvailabilityMode, ThirdPartyBroker, and ThirdPartyBrokerWithLicensing.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ControllerDNSName<String>

Gets sessions that are hosted on machines which are registered with a specific controller.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopGroupName<String>

Gets sessions from a desktop group with the specified name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopGroupUid<Int32>

Gets sessions from a desktop group with the specified UID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopKind<DesktopKind>

Gets sessions on a desktop of a particular kind.

Valid values are Private and Shared.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopSID<String>

Get sessions by desktop SID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopUid<Int32>

Get sessions by desktop Uid.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DeviceId<String>

Get sessions by client device id.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DNSName<String>

Gets sessions by their machine's DNS name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-EstablishmentDuration<Int32>

Gets sessions which took a specific time to establish. In general, Citrix recommends using -Filter and relative comparisons.

Required?	false
Default Value	
Accept Pipeline Input?	false

-EstablishmentTime<DateTime>

Gets sessions which became established at a particular time. In general, Citrix recommends using -Filter and relative comparisons.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HardwareId<String>

Get sessions by client hardware id.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Hidden<Boolean>

Get sessions by whether they are hidden or not. Hidden sessions are treated as though they do not exist when brokering sessions; a hidden session cannot be reconnected to, but a new session may be launched using the same entitlement.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HostedMachineName<String>

Gets sessions by their machine's name as known to its hypervisor.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HostingServerName<String>

Gets sessions hosted by a machine with a specific name of the hosting hypervisor server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HypervisorConnectionName<String>

Gets sessions hosted by a machine with a specific name of the hosting hypervisor connection.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ImageOutOfDate<Boolean>

Gets sessions hosted by a machine with a specific ImageOutOfDate setting.

Required?	false
Default Value	
Accept Pipeline Input?	false

-InMaintenanceMode<Boolean>

Gets sessions hosted by a machine with a specific InMaintenanceMode setting.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IPAddress<String>

Gets sessions hosted by a machine with a specific IP address.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IsAnonymousUser<Boolean>

Gets sessions depending on whether they were established anonymously (\$true) or not (\$false). An anonymous session is established without user credentials and a temporary local user account is used.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IsPhysical<Boolean>

Gets sessions hosted on machines where the flag indicating if the machine can be power managed by the Citrix Broker Service matches the requested value. Where the power state of the machine cannot be controlled, specify \$true, otherwise \$false.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LaunchedViaHostName<String>

Get sessions by the host name of the Web Interface server from which a user launches a session.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LaunchedViaIP<String>

Get sessions by the IP address of the Web Interface server from which a user launches a session.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LogoffInProgress<Boolean>

Gets sessions by whether they are in the process of being logged off or not.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LogonInProgress<Boolean>

Gets sessions by whether they are still executing user logon processing or not.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MachineName<String>

Gets sessions by their machine name (in the form DOMAIN\machine).

Required?	false
Default Value	
Accept Pipeline Input?	false

-MachineSummaryState<DesktopSummaryState>

Gets sessions on a machine with a specific summary state.

Valid values are Off, Unregistered, Available, Disconnected, Preparing, and InUse.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MachineUid<Int32>

Gets sessions on a machine with the specified UID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Metadata<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

Required?	false
Default Value	
Accept Pipeline Input?	false

-OSType<String>

Gets sessions with a specific type of operating system.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PersistUserChanges<PersistUserChanges>

Gets sessions where the user changes are persisted in a particular manner. Values can be:

- o OnLocal - User changes are persisted locally.
- o Discard - User changes are discarded.
- o OnPvd - User changes are persisted on the Pvd.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PowerState<PowerState>

Gets sessions on machines in the specified power state.

Valid values are Unmanaged, Unknown, Unavailable, On, Suspended, TurningOn, TurningOff, Suspending, and Resuming.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Protocol<String>

Get sessions by connection protocol. Valid values are HDX, RDP, or Console.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ProvisioningType<ProvisioningType>

Gets sessions hosted on machines provisioned in a particular manner. Values can be:

- o Manual - No automated provisioning.
- o PVS - Machine provisioned by PVS (machine may be physical, blade, VM,...).
- o MCS - Machine provisioned by MCS (machine must be VM).

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReceiverIPAddress<String>

Gets sessions with the specified client IP address supplied by Receiver (for example, StoreFront) when the session was launched, or reconnected.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReceiverName<String>

Gets sessions with the specified client name supplied by Receiver (for example, StoreFront) when the session was launched, or reconnected.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SecureIcaActive<Boolean>

Get sessions by their use of SecureICA.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionId<Int32>

Deprecated.

Gets sessions by session ID, a unique identifier that Remote Desktop Services uses to track the session but it is only unique on that machine.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionState<SessionState>

Get sessions by their state.

Valid values are Other, PreparingNewSession, Connected, Active, Disconnected, Reconnecting, NonBrokeredSession, and Unknown.

Required?	false

Default Value	
Accept Pipeline Input?	false

-SessionStateChangeTime<DateTime>

Get sessions by their last state change time. In general, Citrix recommends using -Filter and relative comparisons.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionSupport<SessionSupport>

Gets sessions hosted on machines which support the required pattern of sessions. Values can be:

- o SingleSession - Single-session only machine.
- o MultiSession - Multi-session capable machine.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionType<SessionType>

Get sessions by their type.

Valid values are Application and Desktop.

Required?	false
Default Value	
Accept Pipeline Input?	false

-StartTime<DateTime>

Get sessions by their start time. In general, Citrix recommends using -Filter and relative comparisons.

Required?	false
Default Value	
Accept Pipeline Input?	false

-UntrustedUserName<String>

Gets sessions by the untrusted user name reported directly from the machine (in the form DOMAIN\user).

Required?	false
Default Value	
Accept Pipeline Input?	false

-UserFullName<String>

Gets sessions by user's full name (usually 'first-name last-name').

Required?	false
Default Value	
Accept Pipeline Input?	false

-UserName<String>

Get sessions by user name (in the form DOMAIN\user).

Required?	false
Default Value	
Accept Pipeline Input?	false

-UserSID<String>

Get sessions by user's Windows SID.

Required?	false

Default Value	
Accept Pipeline Input?	false

-UserUPN<String>

Gets sessions by user's User Principal Name (in the form user@domain).

Required?	false
Default Value	
Accept Pipeline Input?	false

-ApplicationUid<Int32>

Get sessions running the application with the specified Uid.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SharedDesktopUid<Int32>

Get sessions by SharedDesktop Uid.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about_Broker_Filtering](#) for details.

Required?	false

Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell style filter expression. See about_Broker_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.Session

Returns sessions matching the specified criteria.

Examples

----- **EXAMPLE 1** -----

C:\PS> Get-BrokerSession
Enumerate all sessions.

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerSession -UserName MyDomain\MyAccount -SessionState Disconnected
Get all disconnected sessions for a specific user.
```

Get-BrokerSessionLinger

May 04, 2016

Gets one or more session lingering settings.

Syntax

```
Get-BrokerSessionLinger [-DesktopGroupUid] <Int32> [-Property <String[]>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Get-BrokerSessionLinger [[-DesktopGroupName] <String>] [-AssociatedUserFullName <String>] [-AssociatedUserName  
<String>] [-AssociatedUserUPN <String>] [-Enabled <Boolean>] [-MaxAverageLoadThreshold <Int32>] [-  
MaxLoadPerMachineThreshold <Int32>] [-MaxTimeBeforeDisconnect <TimeSpan>] [-MaxTimeBeforeTerminate  
<TimeSpan>] [-UserFilterEnabled <Boolean>] [-UserSID <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-  
Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Get-BrokerSessionLinger cmdlet is used to enumerate desktop group session linger settings that match all of the supplied criteria.

Without parameters, Get-BrokerSessionLinger gets all the session linger settings that have been created. You can also use the parameters of Get-BrokerSessionLinger to filter the results to just the desktop group you're interested in.

Note that each desktop group can only have a single session linger setting. Session lingering only applies to application sessions.

See about_Broker_Filtering for information about advanced filtering options.

----- BrokerSessionLinger Object

The session linger object returned represents a session linger setting in a desktop group.

-- AssociatedUserFullNames (System.String[])

List of associated users (full names). Associated users is the list of users who are given access using the pre-launch/user mapping filter.

-- AssociatedUserNames (System.String[])

List of associated users (SAM names). Associated users is the list of users who are given access using the pre-launch/user mapping filter.

-- AssociatedUserUPNs (System.String[])

List of associated users (user principle names). Associated users is the list of users who are given access using the pre-launch/user mapping filter.

-- DesktopGroupName (System.String)

Name of the associated desktop group.

-- DesktopGroupUid (System.Int32)

Uid of the associated desktop group.

-- Enabled (System.Boolean)

Specifies whether or not session lingering is enabled for the desktop group.

-- MaxAverageLoadThreshold (System.Int32)

Specifies the average load threshold across the desktop group. After this threshold is hit lingering sessions will be terminated to reduce average load across the group. Sessions that have been lingering the longest will be chosen first.

-- MaxLoadPerMachineThreshold (System.Int32)

Specifies the maximum load threshold per machine in the desktop group. After this threshold is hit lingering sessions on loaded machines will be terminated to reduce load. Sessions that have been lingering the longest will be chosen first.

-- MaxTimeBeforeDisconnect (System.TimeSpan)

Specifies the maximum time by when a lingering session will be disconnected. The disconnect timer cannot be greater than the terminate timer. When the disconnect timer is same as the terminate timer, the session will be directly be terminated. The default value is 15 minutes. A value of 0 disables the disconnect timer.

-- MaxTimeBeforeTerminate (System.TimeSpan)

Specifies the maximum time by when a lingering session will be terminated. When the disconnect timer is same as the terminate timer, the session will be directly be terminated. The default value is 8 hours. A value of 0 disables the terminate timer.

-- UserFilterEnabled (System.Boolean)

Indicates if linger-specific user filter is enabled.

Related topics

[New-BrokerSessionLinger](#)

[Set-BrokerSessionLinger](#)

[Remove-BrokerSessionLinger](#)

Parameters

-DesktopGroupUid<Int32>

Gets session linger setting that is associated with the specified desktop group Uid.

Required?	true
Default Value	
Accept Pipeline Input?	false

-DesktopGroupName<String>

Gets session linger setting that is associated with the specified desktop group name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedUserFullName<String>

Gets session linger settings with an associated user identified by their full name (usually 'first-name last-name'). If the 'UserFilterEnabled' property is true then access to the session linger is restricted to those users only, otherwise access is unrestricted (but always subject to other policy rules).

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedUserName<String>

Gets session linger settings with an associated user identified by their user name (in the form 'domain\user'). If the 'UserFilterEnabled' property is true then access to the session linger is restricted to those users only, otherwise access is unrestricted (but always subject to other policy rules).

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedUserUPN<String>

Gets session linger settings with an associated user identified by their user principle name (in the form 'user@domain'). If the 'UserFilterEnabled' property is true then access to the session linger is restricted to those users only, otherwise access is unrestricted (but always subject to other policy rules).

Required?	false

Default Value	
Accept Pipeline Input?	false

-Enabled<Boolean>

Gets only the session linger settings that have the specified value for whether the setting is enabled.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MaxAverageLoadThreshold<Int32>

Gets only the session linger settings that have the specified average load threshold.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MaxLoadPerMachineThreshold<Int32>

Gets only the session linger settings that have the specified maximum load threshold per machine.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MaxTimeBeforeDisconnect<TimeSpan>

Gets only the session linger settings that have the specified idle disconnect time.

Required?	false
Default Value	

Accept Pipeline Input?	false

-MaxTimeBeforeTerminate<TimeSpan>

Gets only the session linger settings that have the specified idle terminate time.

Required?	false
Default Value	
Accept Pipeline Input?	false

-UserFilterEnabled<Boolean>

Gets only session linger settings whose user filter is in the specified state.

Required?	false
Default Value	
Accept Pipeline Input?	false

-UserSID<String>

Gets only session linger settings with their accessibility restricted to include the specified user.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about_Broker_Filtering](#) for details.

Required?	false
Default Value	False

Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell style filter expression. See about_Broker_Filtering for details.

Required?	false

Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.SessionLinger

Get-BrokerSessionLinger returns an object for each session linger setting it gets.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerSessionLinger -DesktopGroupName "test"
```

Returns the session linger settings associated with the desktop group named 'test'.

Get-BrokerSessionPreLaunch

May 04, 2016

Gets one or more session pre-launch settings.

Syntax

```
Get-BrokerSessionPreLaunch [-DesktopGroupUid] <Int32> [-Property <String[]>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Get-BrokerSessionPreLaunch [[-DesktopGroupName] <String>] [-AssociatedUserFullName <String>] [-AssociatedUserName  
<String>] [-AssociatedUserUPN <String>] [-Enabled <Boolean>] [-MaxAverageLoadThreshold <Int32>] [-  
MaxLoadPerMachineThreshold <Int32>] [-MaxTimeBeforeDisconnect <TimeSpan>] [-MaxTimeBeforeTerminate  
<TimeSpan>] [-UserFilterEnabled <Boolean>] [-UserSID <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-  
Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Get-BrokerSessionPreLaunch cmdlet is used to enumerate desktop group session pre-launch settings that match all of the supplied criteria.

Without parameters, Get-BrokerSessionPreLaunch gets all the session pre-launch settings that have been created. You can also use the parameters of Get-BrokerSessionPreLaunch to filter the results to just the desktop group you're interested in.

Note that each desktop group can only have a single session pre-launch setting. Session pre-launch only applies to application sessions.

See about_Broker_Filtering for information about advanced filtering options.

----- BrokerSessionPreLaunch Object

The session pre-launch object returned represents a session pre-launch setting in a desktop group.

-- AssociatedUserFullNames (System.String[])

List of associated users (full names). Associated users is the list of users who are given access using the pre-launch/user mapping filter.

-- AssociatedUserNames (System.String[])

List of associated users (SAM names). Associated users is the list of users who are given access using the pre-launch/user mapping filter.

-- AssociatedUserUPNs (System.String[])

List of associated users (user principle names). Associated users is the list of users who are given access using the pre-launch/user mapping filter.

-- DesktopGroupName (System.String)

Name of the associated desktop group.

-- DesktopGroupUid (System.Int32)

Uid of the associated desktop group.

-- Enabled (System.Boolean)

Specifies whether or not session pre-launch is enabled for the desktop group.

-- MaxAverageLoadThreshold (System.Int32)

Specifies the average load threshold across the desktop group. After this threshold is hit pre-launched sessions will be terminated to reduce average load across the group. Sessions that have been pre-launched the longest will be chosen first.

-- MaxLoadPerMachineThreshold (System.Int32)

Specifies the maximum load threshold per machine in the desktop group. After this threshold is hit pre-launched sessions on loaded machines will be terminated to reduce load. Sessions that have been pre-launched the longest will be chosen first.

-- MaxTimeBeforeDisconnect (System.TimeSpan)

Specifies the maximum time by when a pre-launched session will be disconnected. The disconnect timer cannot be greater than the terminate timer. When the disconnect timer is same as the terminate timer, the session will be directly be terminated. The default value is 15 minutes. A value of 0 disables the disconnect timer.

-- MaxTimeBeforeTerminate (System.TimeSpan)

Specifies the maximum time by when a pre-launched session will be terminated. When the disconnect timer is same as the terminate timer, the session will be directly be terminated. The default value is 8 hours. A value of 0 disables the terminate timer.

-- UserFilterEnabled (System.Boolean)

Indicates if pre-launch-specific user filter is enabled.

Related topics

[New-BrokerSessionPreLaunch](#)

[Set-BrokerSessionPreLaunch](#)

[Remove-BrokerSessionPreLaunch](#)

Parameters

-DesktopGroupUid<Int32>

Gets session pre-launch setting that is associated with the specified desktop group Uid.

Required?	true
Default Value	
Accept Pipeline Input?	false

-DesktopGroupName<String>

Gets session pre-launch setting that is associated with the specified desktop group name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedUserFullName<String>

Gets session pre-launch settings with an associated user identified by their full name (usually 'first-name last-name'). If the 'UserFilterEnabled' property is true then access to the session pre-launch is restricted to those users only, otherwise access is unrestricted (but always subject to other policy rules).

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedUserName<String>

Gets session pre-launch settings with an associated user identified by their user name (in the form 'domain\user'). If the 'UserFilterEnabled' property is true then access to the session pre-launch is restricted to those users only, otherwise access is unrestricted (but always subject to other policy rules).

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedUserUPN<String>

Gets session pre-launch settings with an associated user identified by their user principle name (in the form 'user@domain'). If the 'UserFilterEnabled' property is true then access to the session pre-launch is restricted to those users only, otherwise access is unrestricted (but always subject to other policy rules).

Required?	false

Default Value	
Accept Pipeline Input?	false

-Enabled<Boolean>

Gets only the session pre-launch settings that have the specified value for whether the setting is enabled.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MaxAverageLoadThreshold<Int32>

Gets only the session pre-launch settings that have the specified average load threshold.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MaxLoadPerMachineThreshold<Int32>

Gets only the session pre-launch settings that have the specified maximum load threshold per machine.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MaxTimeBeforeDisconnect<TimeSpan>

Gets only the session pre-launch settings that have the specified idle disconnect time.

Required?	false
Default Value	

Accept Pipeline Input?	false

-MaxTimeBeforeTerminate<TimeSpan>

Gets only the session pre-launch settings that have the specified idle terminate time.

Required?	false
Default Value	
Accept Pipeline Input?	false

-UserFilterEnabled<Boolean>

Gets only session pre-launch settings whose user filter is in the specified state.

Required?	false
Default Value	
Accept Pipeline Input?	false

-UserSID<String>

Gets only session pre-launch settings with their accessibility restricted to include the specified user.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about_Broker_Filtering](#) for details.

Required?	false
Default Value	False

Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell style filter expression. See about_Broker_Filtering for details.

Required?	false

Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.SessionPreLaunch

Get-BrokerSessionPreLaunch returns an object for each session pre-launch setting it gets.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerSessionPreLaunch -DesktopGroupName "test"
```

Returns the session pre-launch settings associated with the destkop group named 'test'.

Get-BrokerSharedDesktop

May 04, 2016

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

May 04, 2016

Gets the current XenDesktop broker site.

Syntax

```
Get-BrokerSite [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Get-BrokerSite cmdlet gets the current broker site.

The broker site is a top-level, logical representation of the XenDesktop site, from the perspective of the brokering services running within the site. It defines various site-wide default attributes used by the brokering services.

A XenDesktop installation has only a single broker site instance.

----- BrokerSite Object

The BrokerSite object represents logical representation of the XenDesktop site. It contains the following properties:

-- BaseOU (System.Guid?)

The objectGUID property identifying the base OU in Active Directory used for desktop registrations.

-- BrokerServiceGroupUid (System.Guid)

The Uid for the Broker Service Group.

-- ColorDepth (Citrix.Broker.Admin.SDK.ColorDepth)

The default color depth for new desktop groups.

-- ConfigurationServiceGroupUid (System.Guid?)

The Uid for the Configuration Service Group.

-- ConnectionLeasingEnabled (System.Boolean?)

The indicator for connection leasing active.

-- DesktopGroupIconUid (System.Int32)

The default desktop icon used for new desktop groups.

-- DnsResolutionEnabled (System.Boolean)

The setting to configure whether numeric IP address or the DNS name to be present in the ICA file.

-- LicensedSessionsActive (System.Int32?)

The count of active licensed session.

-- LicenseEdition (System.String)

The license edition for session brokering.

-- LicenseGraceSessionsRemaining (System.Int32?)

The count of License Grace Session Remaining

-- LicenseModel (Citrix.Broker.Admin.SDK.LicenseModel?)

The licensing model in use. Values can be 'Concurrent' or 'UserDevice'

-- LicenseServerName (System.String)

The DNS for License Server Name

-- LicenseServerPort (System.Int32)

The port for the License Server

-- LicensingBurnIn (System.String)

The date for the license to end in yyyy.MMdd format

-- LicensingBurnInDate (System.DateTime?)

The date for the license to end

-- LicensingGraceHoursLeft (System.Int32?)

The number of grace hours left after license expiry

-- LicensingGracePeriodActive (System.Boolean?)

The indicator for licensing grace period active

-- LicensingOutOfBoxGracePeriodActive (System.Boolean?)

The indicator for licensing out of the box grace period active

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

The metadata for this command.

-- Name (System.String)

The name of the site

-- SecureIcaRequired (System.Boolean)

The default SecureICA usage requirements for new desktop groups.

-- TrustRequestsSentToTheXmlServicePort (System.Boolean)

The XML Service trust settings.

Related topics

[Set-BrokerSite](#)

[Get-BrokerIcon](#)

Parameters

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.Site

Get-BrokerSite returns the single broker site instance.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerSite
```

Gets the current broker site.

Get-BrokerTag

May 04, 2016

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

May 04, 2016

Gets machines that have registered but are not yet configured in this site.

Syntax

```
Get-BrokerUnconfiguredMachine -SID <String> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerUnconfiguredMachine [[-MachineName] <String>] [-AgentVersion <String>] [-ControllerDNSName <String>] [-DNSName <String>] [-FunctionalLevel <FunctionalLevel>] [-LastDeregistrationTime <DateTime>] [-OSType <String>] [-OSVersion <String>] [-SessionSupport <SessionSupport>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Retrieve machines matching the specified criteria, where the machine has registered with a controller in the site but the machine has not yet been configured to be part of the site. If no parameters are specified, this cmdlet enumerates all such machines.

See about_Broker_Filtering for information about advanced filtering options, and about_Broker_Machines for background information about machines.

----- BrokerUnconfiguredMachine Object

An unconfigured machine is a machine that has registered with the site but is not configured in either a desktop group or a catalog.

-- AgentVersion (System.String)

Version of the Citrix Virtual Delivery Agent (VDA) installed on the unconfigured machine.

-- ControllerDNSName (System.String)

The DNS name of the controller that the unconfigured machine is registered with.

-- DNSName (System.String)

The DNS name of the unconfigured machine.

-- FunctionalLevel (Citrix.Broker.Admin.SDK.FunctionalLevel?)

The functional level of the unconfigured machine. This is determined by the version of the Citrix VDA software installed on the machine.

-- LastDeregistrationTime (System.DateTime?)

The time when the unconfigured machine last deregistered with the Citrix Broker Service.

-- MachineName (System.String)

The machine name of the unconfigured machine in the form domain\machine.

-- OSType (System.String)

The type of operating system installed on the unconfigured machine.

-- OSVersion (System.String)

The version of the operating system installed on the unconfigured machine.

-- SessionSupport (Citrix.Broker.Admin.SDK.SessionSupport?)

The session support of the unconfigured machine. Valid values are:

SingleSession, MultiSession

-- SID (System.String)

Security identifier of the unconfigured machine.

Related topics

[Get-BrokerMachine](#)

Parameters

-SID<String>

Gets machines by their machine SID.

Required?	true
Default Value	
Accept Pipeline Input?	false

-MachineName<String>

Gets machines by their machine name (in the form domain\machine).

Required?	false
Default Value	
Accept Pipeline Input?	false

-AgentVersion<String>

Gets machines with a specific Virtual Desktop Agent version.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ControllerDNSName<String>

Gets machines by the DNS name of the controller they are registered with.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DNSName<String>

Gets machines by their DNS name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-FunctionalLevel<FunctionalLevel>

Gets machines with a specific FunctionalLevel.

Valid values are L5, L7, L7_6

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastDeregistrationTime<DateTime>

Gets machines by the time that they were last deregistered.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OSType<String>

Gets machines by the type of operating system they are running.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OSVersion<String>

Gets machines by the version of the operating system they are running.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionSupport<SessionSupport>

Gets machines that have the specified session capability. Values can be:

- o SingleSession - Single-session only machine.
- o MultiSession - Multi-session capable machine.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about_Broker_Filtering for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell style filter expression. See [about_Broker_Filtering](#) for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through `Select-Object`, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.UnconfiguredMachine

Get-BrokerUnconfiguredMachine returns an object for each matching machine

Notes

It is generally better to compare dates and times using `-Filter` and relative comparisons. See [about_Broker_Filtering](#) and the examples in this topic for more information.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Get-BrokerUnconfiguredMachine -Filter { ControllerDNSName -eq 'Controller1.domain.com' -and OSType -eq 'Windows XP Service Pack 3' }
```

This command returns all unconfigured XP SP3 machines which are registered with the controller called 'Controller1.domain.com'.

Get-BrokerUser

May 04, 2016

Gets broker users configured for this site.

Syntax

```
Get-BrokerUser -SID <String> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerUser [[-Name] <String>] [-FullName <String>] [-UPN <String>] [-ApplicationUid <Int32>] [-SessionLingerDesktopGroupUid <Int32>] [-SessionPreLaunchDesktopGroupUid <Int32>] [-MachineUid <Int32>] [-PrivateDesktopUid <Int32>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Retrieve broker users matching the specified criteria. If no parameters are specified this cmdlet enumerates all broker users.

For information about advanced filtering options, see [about_Broker_Filtering](#).

----- BrokerUser Object

The BrokerUser object represents a single instance of an user. It contains the following properties:

-- FullName (System.String)

The full name of an user

-- Name (System.String)

The name of an user

-- SID (System.String)

The SID of an user

-- UPN (System.String)

The UPN of an user

Related topics

[Add-BrokerUser](#)

[Remove-BrokerUser](#)

Parameters

-SID<String>

Gets the broker user with the specified SID property value.

Required?	true

Default Value	
Accept Pipeline Input?	false

-Name<String>

Gets the broker user with the specified Name property.

Required?	false
Default Value	
Accept Pipeline Input?	false

-FullName<String>

Gets the broker user with the specified FullName property.

Required?	false
Default Value	
Accept Pipeline Input?	false

-UPN<String>

Gets the broker user with the specified UPN property value.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ApplicationUid<Int32>

Gets broker users associated with the application with the specified Uid.

Required?	false
Default Value	

Accept Pipeline Input?	false

-SessionLingerDesktopGroupUid<Int32>

Gets broker users associated with the desktop group session linger settings with the specified Uid.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionPreLaunchDesktopGroupUid<Int32>

Gets broker users associated with the desktop group session pre-launch settings with the specified Uid.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MachineUid<Int32>

Gets broker users associated with the broker machine with the specified Uid.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PrivateDesktopUid<Int32>

Gets broker users associated with the private desktop with the specified Uid.

Required?	false
Default Value	

Accept Pipeline Input?	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

May 04, 2016

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.q

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedUserFullName<String>

Gets desktops with an associated user identified by their full name (usually in the form 'first-name last-name').

Associated users are the current user for shared desktops, and the assigned users for private desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedUserName<String>

Gets desktops with an associated user identified by their user name (in the form 'domain\user').

Associated users are the current user for shared desktops, and the assigned users for private desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedUserUPN<String>

Gets desktops with an associated user identified by their User Principle Name (in the form 'user@domain').

Associated users are the current user for shared desktops, and the assigned users for private desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AutonomouslyBrokered<Boolean>

Gets desktops according to whether their current session is autonomously brokered or not. Autonomously brokered sessions are HDX sessions established by

direct connection without being brokered.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-CatalogName<String>

Gets desktops from the catalog with the specific name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-CatalogUid<Int32>

Gets desktops from a catalog with a specific UID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ClientAddress<String>

Gets desktops with a specific client IP address.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ClientName<String>

Gets desktops with a specific client name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ClientVersion<String>

Gets desktops with a specific client version.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ColorDepth<ColorDepth>

Gets desktops configured with a specific color depth.

Valid values are FourBit, EightBit, SixteenBit, and TwentyFourBit.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ConnectedViaHostName<String>

Gets desktops with a specific host name of the incoming connection. This is usually a proxy or Citrix Access Gateway server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ConnectedViaIP<String>

Gets desktops with a specific IP address of the incoming connection.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ControllerDNSName<String>

Gets desktops with a specific DNS name of the controller they are registered with.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DeliveryType<DeliveryType>

Gets desktops of a particular delivery type.

Valid values are AppsOnly, DesktopsOnly, DesktopsAndApps

Required?	false

Default Value	
Accept Pipeline Input?	false

-Description<String>

Get desktops with a specific description.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopCondition<String>

Gets desktop with an outstanding desktop condition condition.

Valid values are:

- o CPU: Indicates the machine has high CPU usage
- o ICALatency: Indicates the network latency is high
- o UPMLogonTime: Indicates that the profile load time was high

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopGroupName<String>

Gets desktops from a desktop group with the specified name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopGroupUid<Int32>

Gets desktops from a desktop group with the specified UID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopKind<DesktopKind>

Deprecated: Use AllocationType parameter.

Gets desktops of a particular kind.

Valid values are Private, Shared.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DeviceId<String>

Gets desktops with a specific client device ID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DNSName<String>

Get desktops with a specific DNS name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-FunctionalLevel<FunctionalLevel>

Gets desktops with a specific FunctionalLevel.

Valid values are L5, L7, L7_6

Required?	false
Default Value	
Accept Pipeline Input?	false

-HardwareId<String>

Gets desktops with a specific client hardware ID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HostedMachineId<String>

Gets desktops with a specific machine ID known to the hypervisor.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HostedMachineName<String>

Gets desktops with a specific machine name known to the hypervisor.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HostingServerName<String>

Gets desktops with a specific name of the hosting hypervisor server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HypervisorConnectionName<String>

Gets desktops with a specific name of the hosting hypervisor connection.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HypervisorConnectionUid<Int32>

Gets desktops with a specific UID of the hosting hypervisor connection.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IconUid<Int32>

Gets desktops with a specific configured icon. Note that desktops with a null IconUid use the icon of the desktop group.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ImageOutOfDate<Boolean>

Gets desktops if they have an ImageOutOfDate flag.

Required?	false
Default Value	
Accept Pipeline Input?	false

-InMaintenanceMode<Boolean>

Gets desktops with a specific InMaintenanceMode setting.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IPAddress<String>

Gets desktops with a specific IP address.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IsAssigned<Boolean>

Gets desktops according to whether they are assigned or not. Desktops may be assigned to one or more users or groups, a client IP address or a client endpoint name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IsPhysical<Boolean>

Specifies if machines in the catalog can be power managed by the Citrix Broker Service. Where the power state of the machine cannot be controlled, specify \$true, otherwise \$false. Can only be specified together with a provisioning type of Pvs or Manual, or if used with the deprecated CatalogKind parameter only with Pvs or PvsPvd catalog kinds.

Required?	false

Default Value	
Accept Pipeline Input?	false

-LastConnectionFailure<ConnectionFailureReason>

Gets desktops with a specific reason for the last recorded connection failure. This value is None if the last connection was successful or if there has been no attempt to connect to the desktop yet.

Valid values are None, SessionPreparation, RegistrationTimeout, ConnectionTimeout, Licensing, Ticketing, and Other.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastConnectionTime<DateTime>

Gets desktops that last connected at a specific time. This is the time that the broker detected that the connection attempt either succeeded or failed.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastConnectionUser<String>

Gets desktops where a specific user name last attempted a connection (in the form 'domain\user').

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastDeregistrationReason<DeregistrationReason>

Gets desktops whose broker last recorded a specific deregistration reason.

Valid values are \$null, AgentShutdown, AgentSuspended, AgentRequested, IncompatibleVersion, AgentAddressResolutionFailed, AgentNotContactable, AgentWrongActiveDirectoryOU, EmptyRegistrationRequest, MissingRegistrationCapabilities, MissingAgentVersion, InconsistentRegistrationCapabilities, NotLicensedForFeature, UnsupportedCredentialSecurityVersion, InvalidRegistrationRequest, SingleMultiSessionMismatch, FunctionalLevelTooLowForCatalog, FunctionalLevelTooLowForDesktopGroup, PowerOff, DesktopRestart, DesktopRemoved, AgentRejectedSettingsUpdate, SendSettingsFailure, SessionAuditFailure, SessionPrepareFailure, ContactLost, SettingsCreationFailure, UnknownError and BrokerRegistrationLimitReached.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastDeregistrationTime<DateTime>

Gets desktops that were last deregistered by a specific time.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastErrorReason<String>

Gets desktops with the specified last error reason.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastErrorTime<DateTime>

Gets desktops with the specified last error time.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastHostingUpdateTime<DateTime>

Gets desktops with a specific time that the hosting information was last updated.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LaunchedViaHostName<String>

Gets desktops with a specific host name of the StoreFront server from which the user launched the session.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LaunchedViaIP<String>

Gets desktops with a specific IP address of the StoreFront server from which the user launched the session.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MachineInternalState<MachineInternalState>

Gets desktops with the specified internal machine state.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MachineUid<Int32>

Gets desktops with a specific machine UID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OSType<String>

Gets desktops by the type of operating system they are running.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OSVersion<String>

Gets desktops by the version of the operating system they are running.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PersistUserChanges<PersistUserChanges>

Gets desktops by the location where the user changes are persisted.

- o OnLocal - User changes are persisted locally.
- o Discard - User changes are discarded.
- o OnPvd - User changes are persisted on the Pvd.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PowerActionPending<Boolean>

Gets desktops with a specific power action pending state.

Valid values are \$true or \$false.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PowerState<PowerState>

Gets desktops with a specific power state.

Valid values are Unmanaged, Unknown, Unavailable, Off, On, Suspended, TurningOn, TurningOff, Suspending, and Resuming.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Protocol<String>

Gets desktops with connections using a specific protocol, for example HDX, RDP, or Console.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ProvisioningType<ProvisioningType>

Specifies the provisioning type for the catalog. Values can be:

- o Manual - No provisioning.
- o PVS - Machine provisioned by PVS (machine may be physical, blade, VM,...).
- o MCS - Machine provisioned by MCS (machine must be VM).

Required?	false

Default Value	
Accept Pipeline Input?	false

-PublishedApplication<String>

Gets desktops with a specific application published on them (identified by its browser name).

Required?	false
Default Value	
Accept Pipeline Input?	false

-PublishedName<String>

Gets desktops with a specific published name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PvdStage<PvdStage>

Gets machines with a specific personal vDisk stage.

Valid values are None, Requested, Starting, Working and Failed.

Required?	false
Default Value	
Accept Pipeline Input?	false

-RegistrationState<RegistrationState>

Gets desktops with a specific registration state.

Valid values are Unregistered, Initializing, Registered and AgentError.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SecureIcaActive<Boolean>

Gets desktops depending on whether the current session uses SecureICA or not.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SecurelcaRequired<Boolean>

Gets desktops configured with a particular SecurelcaRequired setting. Note that the desktop setting of \$null indicates that the desktop group value is used.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionHidden<Boolean>

Gets desktops by whether their sessions are hidden or not. Hidden sessions are treated as though they do not exist when launching sessions; a hidden session cannot be reconnected to, but a new session may be launched using the same entitlement.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionId<Int32>

Deprecated.

Gets desktops by session ID, a unique identifier that Remote Desktop Services uses to track the session but it is only unique on that machine.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionState<SessionState>

Gets desktops with a specific session state.

Valid values are \$null, Other, PreparingSession, Connected, Active, Disconnected, Reconnecting, NonBrokeredSession, and Unknown.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionStateChangeTime<DateTime>

Gets desktops whose sessions last changed state at a specific time.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionUid<Int64>

Gets desktops with a specific session UID (\$null for no session).

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionUserName<String>

Gets desktops with a specific user name for the current session (in the form 'domain\user').

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionUserSID<String>

Gets desktops with a specific SID of the current session user.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SID<String>

Gets desktops with a specific machine SID.

Required?	false
Default Value	

Accept Pipeline Input?	false

-SmartAccessTag<String>

Gets desktops where the session has the specific SmartAccess tag.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-StartTime<DateTime>

Gets desktops with a specific session start time.

Session properties are always null for multi-session desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SummaryState<DesktopSummaryState>

Gets desktops with a specific summary state.

Valid values are Off, Unregistered, Available, Disconnected, and InUse.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Tag<String>

Gets desktops with a specific tag.

Required?	false
Default Value	
Accept Pipeline Input?	false

-WillShutdownAfterUse<Boolean>

Gets desktops depending on whether they shut down after use or not.

Required?	false

Default Value	
Accept Pipeline Input?	false

-ApplicationUid<Int32>

Gets desktops with a specific published application (identified by its UID).

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about_Broker_Filtering for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.

Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell style filter expression. See [about_Broker_Filtering](#) for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.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

May 04, 2016

Groups and counts machines with the same value for a specified property.

Syntax

```
Group-BrokerMachine [-UId <Int32> -Property <String> [-AdminAddress <String>] [<CommonParameters>]
```

```
Group-BrokerMachine -Property <String> [[-MachineName <String>] [-AgentVersion <String>] [-AllocationType <AllocationType>] [-ApplicationInUse <String>] [-AssignedClientName <String>] [-AssignedIPAddress <String>] [-AssociatedUserFullName <String>] [-AssociatedUserName <String>] [-AssociatedUserSID <String>] [-AssociatedUserUPN <String>] [-BrowserName <String>] [-CatalogName <String>] [-CatalogUId <Int32>] [-CatalogUUID <Guid>] [-ColorDepth <ColorDepth>] [-ControllerDNSName <String>] [-DeliveryType <DeliveryType>] [-Description <String>] [-DesktopCondition <String>] [-DesktopGroupName <String>] [-DesktopGroupUId <Int32>] [-DesktopGroupUUID <Guid>] [-DesktopKind <DesktopKind>] [-DesktopUId <Int32>] [-DNSName <String>] [-FaultState <MachineFaultState>] [-FunctionalLevel <FunctionalLevel>] [-HostedMachineId <String>] [-HostedMachineName <String>] [-HostingServerName <String>] [-HypervisorConnectionName <String>] [-HypervisorConnectionUId <Int32>] [-HypHypervisorConnectionUId <Guid>] [-IconUId <Int32>] [-ImageOutOfDate <Boolean>] [-InMaintenanceMode <Boolean>] [-IPAddress <String>] [-IsAssigned <Boolean>] [-IsPhysical <Boolean>] [-LastConnectionFailure <ConnectionFailureReason>] [-LastConnectionTime <DateTime>] [-LastConnectionUser <String>] [-LastDeregistrationReason <DeregistrationReason>] [-LastDeregistrationTime <DateTime>] [-LastErrorReason <String>] [-LastErrorTime <DateTime>] [-LastHostingUpdateTime <DateTime>] [-LastPvdErrorReason <String>] [-LastPvdErrorTime <DateTime>] [-LoadIndex <Int32>] [-MachineInternalState <MachineInternalState>] [-Metadata <String>] [-OSType <String>] [-OSVersion <String>] [-PersistUserChanges <PersistUserChanges>] [-PowerActionPending <Boolean>] [-PowerState <PowerState>] [-ProvisioningType <ProvisioningType>] [-PublishedApplication <String>] [-PublishedName <String>] [-PvdEstimatedCompletionTime <DateTime>] [-PvdPercentDone <Int32>] [-PvdStage <PvdStage>] [-PvdUpdateStartTime <DateTime>] [-RegistrationState <RegistrationState>] [-ScheduledReboot <ScheduledReboot>] [-SecureIcaRequired <Boolean>] [-SessionAutonomouslyBrokered <Boolean>] [-SessionClientAddress <String>] [-SessionClientName <String>] [-SessionClientVersion <String>] [-SessionConnectedViaHostName <String>] [-SessionConnectedViaIP <String>] [-SessionCount <Int32>] [-SessionDeviceId <String>] [-SessionHardwareId <String>] [-SessionHidden <Boolean>] [-SessionKey <Guid>] [-SessionLaunchedViaHostName <String>] [-SessionLaunchedViaIP <String>] [-SessionProtocol <String>] [-SessionSecureIcaActive <Boolean>] [-SessionsEstablished <Int32>] [-SessionSmartAccessTag <String>] [-SessionsPending <Int32>] [-SessionStartTime <DateTime>] [-SessionState <SessionState>] [-SessionStateChangeTime <DateTime>] [-SessionSupport <SessionSupport>] [-SessionType <SessionType>] [-SessionUId <Int64>] [-SessionUserName <String>] [-SessionUserSID <String>] [-SID <String>] [-SummaryState <DesktopSummaryState>] [-SupportedPowerActions <String[]>] [-Tag <String>] [-UUID <Guid>] [-VMToolsState <VMToolsState>] [-WillShutdownAfterUse <Boolean>] [-WindowsConnectionSetting <WindowsConnectionSetting>] [-AssignedUserSID <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Filters machines using the specified criteria, then groups and counts matching machines with the same value for a particular property. The number of machines in the group, and the property value for the group, is output. For example:

```
C:\PS> Group-BrokerMachine -Property SummaryState
```

```
Count Name
```

```
-----
```

```
43 Available
```

```
17 InUse
```

```
3 Disconnected
```

Filtering supports the same options as the Get-BrokerMachine cmdlet, and allows filtering on both machine and session properties.

Group-BrokerMachine is similar to the standard PowerShell Group-Object, but is faster than piping the output of Get-BrokerMachine into Group-Object when working with many machines.

Note that the MaxRecordCount, ReturnTotalRecordCount, Skip, and SortBy parameters apply to GroupInfo records output rather than the filtered machines.

Related topics

[Get-BrokerMachine](#)

[Group-Object](#)

[Parameters](#)

-UId<Int32>

Gets a machine with a specific UID.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Property<String>

Selects the property by which matching machines are grouped.

Required?	true
Default Value	
Accept Pipeline Input?	false

-MachineName<String>

Gets machines with a specific machine name (in the form domain\machine).

Required?	false
Default Value	
Accept Pipeline Input?	false

-AgentVersion<String>

Gets machines with a specific Virtual Delivery Agent version.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AllocationType<AllocationType>

Gets machines from catalogs with the specified allocation type.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ApplicationInUse<String>

Gets machines running a specified published application. String comparisons are case-insensitive.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AssignedClientName<String>

Gets machines that have been assigned to the specific client name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssignedIPAddress<String>

Gets machines that have been assigned to the specific client IP address.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedUserFullName<String>

Gets machines with an associated user identified by their full name (usually 'first-name last-name').

Associated users are all current users of a desktop, plus the assigned users for private desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedUserName<String>

Gets machines with an associated user identified by their user name (in the form 'domain\user').

Associated users are all current users of a desktop, plus the assigned users for private desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssociatedUserSID<String>

Gets machines with an associated user identified by their Windows SID.

Associated users are all current users of a desktop, plus the assigned users for private desktops.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AssociatedUserUPN<String>

Gets machines with an associated user identified by their User Principle Name (in the form 'user@domain').

Associated users are all current users of a desktop, plus the assigned users for private desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-BrowserName<String>

Gets assigned machines backing desktop resources that have browser names matching the specified name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-CatalogName<String>

Gets machines from the catalog with the specific name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-CatalogUid<Int32>

Gets machines from the catalog with the specific UID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-CatalogUUID<Guid>

Gets machines from the catalog with the specific UUID.

Required?	false
Default Value	

Accept Pipeline Input?	false

-ColorDepth<ColorDepth>

Gets machines configured with a specific color depth.

Valid values are FourBit, EightBit, SixteenBit, and TwentyFourBit.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ControllerDNSName<String>

Gets machines by the DNS name of the controller they are registered with.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DeliveryType<DeliveryType>

Gets machines of a particular delivery type.

Valid values are AppsOnly, DesktopsOnly, DesktopsAndApps

Required?	false
Default Value	
Accept Pipeline Input?	false

-Description<String>

Get machines by description.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopCondition<String>

Gets machines with an outstanding desktop condition.

Valid values are:

- o CPU: Indicates the machine has high CPU usage
- o ICALatency: Indicates the network latency is high

o UPMLogonTime: Indicates that the profile load time was high

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopGroupName<String>

Gets machines from a desktop group with the specified name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopGroupUid<Int32>

Gets machines from a desktop group with a specific UID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopGroupUUID<Guid>

Gets machines from a desktop group with a specific UUID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopKind<DesktopKind>

Deprecated: Use AllocationType parameter.

Gets machines of a particular kind.

Valid values are Private, Shared.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopUid<Int32>

Gets the machine that corresponds to the desktop with the specific UID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DNSName<String>

Gets machines with the specific DNS name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-FaultState<MachineFaultState>

Gets machines currently in the specified fault state.

Required?	false
Default Value	
Accept Pipeline Input?	false

-FunctionalLevel<FunctionalLevel>

Gets machines with a specific FunctionalLevel.

Valid values are L5, L7, L7_6

Required?	false
Default Value	
Accept Pipeline Input?	false

-HostedMachineId<String>

Gets machines with the specific machine ID known to the hypervisor.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HostedMachineName<String>

Gets machines with the specific machine name known to the hypervisor.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HostingServerName<String>

Gets machines by the name of the hosting hypervisor server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HypervisorConnectionName<String>

Gets machines with the specific name of the hypervisor connection hosting them.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HypervisorConnectionUid<Int32>

Gets machines with the specific UID of the hypervisor connection hosting them.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HypHypervisorConnectionUid<Guid>

Gets machines with the specific UUID of the hypervisor connection hosting them.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IconUid<Int32>

Gets machines by configured icon. Note that machines with a null IconUid use the icon of the desktop group.

Required?	false
Default Value	

Accept Pipeline Input?	false

-ImageOutOfDate<Boolean>

Gets machines depending on whether their disk image is out of date or not (for machines provisioned using MCS only).

Required?	false
Default Value	
Accept Pipeline Input?	false

-InMaintenanceMode<Boolean>

Gets machines by whether they are in maintenance mode or not.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IPAddress<String>

Gets machines with a specific IP address.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IsAssigned<Boolean>

Gets machines according to whether they are assigned or not. Machines may be assigned to one or more users or groups, a client IP address or a client endpoint name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IsPhysical<Boolean>

Gets machines according to whether they can be power managed by XenDesktop or not.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastConnectionFailure<ConnectionFailureReason>

Gets machines with a specific reason for the last recorded connection failure. This value is None if the last connection was successful or if there has been no attempt to connect to the machine yet.

Valid values are None, SessionPreparation, RegistrationTimeout, ConnectionTimeout, Licensing, Ticketing, and Other.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastConnectionTime<DateTime>

Gets machines to which a user session connection occurred at a specific time. This is the time that the broker detected that the connection attempt either succeeded or failed.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastConnectionUser<String>

Gets machines where a specific user name last attempted a connection (in the form 'domain\user').

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastDeregistrationReason<DeregistrationReason>

Gets machines whose broker last recorded a specific deregistration reason.

Valid values are Null, AgentShutdown, AgentSuspended, AgentRequested, IncompatibleVersion, AgentAddressResolutionFailed, AgentNotContactable, AgentWrongActiveDirectoryOU, EmptyRegistrationRequest, MissingRegistrationCapabilities, MissingAgentVersion, InconsistentRegistrationCapabilities, NotLicensedForFeature, UnsupportedCredentialSecurityVersion, InvalidRegistrationRequest, SingleMultiSessionMismatch, FunctionalLevelTooLowForCatalog, FunctionalLevelTooLowForDesktopGroup, PowerOff, DesktopRestart, DesktopRemoved, AgentRejectedSettingsUpdate, SendSettingsFailure, SessionAuditFailure, SessionPrepareFailure, ContactLost, SettingsCreationFailure, UnknownError and BrokerRegistrationLimitReached.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastDeregistrationTime<DateTime>

Gets machines by the time that they were last deregistered.

Required?	false

Default Value	
Accept Pipeline Input?	false

-LastErrorReason<String>

Gets machines with the specified last error reason.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastErrorTime<DateTime>

Gets machines with the specified last error time.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastHostingUpdateTime<DateTime>

Gets machines with a specific time that the hosting information was last updated.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastPvdErrorReason<String>

Gets machines with the specified last Personal vDisk preparation error reason.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LastPvdErrorTime<DateTime>

Gets machines with the specified last Personal vDisk preparation error time.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LoadIndex<Int32>

Gets machines by their current load index.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MachineInternalState<MachineInternalState>

Gets machines with the specified internal state.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Metadata<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

Required?	false
Default Value	
Accept Pipeline Input?	false

-OSType<String>

Gets machines by the type of operating system they are running.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OSVersion<String>

Gets machines by the version of the operating system they are running.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PersistUserChanges<PersistUserChanges>

Gets machines according to the location where user changes are persisted. Values can be:

- o OnLocal - User changes are persisted locally.
- o Discard - User changes are discarded.
- o OnPvd - User changes are persisted on the Pvd.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PowerActionPending<Boolean>

Gets machines depending on whether a power action is pending or not.

Valid values are \$true or \$false.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PowerState<PowerState>

Gets machines with a specific power state.

Valid values are Unmanaged, Unknown, Unavailable, Off, On, Suspended, TurningOn, TurningOff, Suspending, and Resuming.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ProvisioningType<ProvisioningType>

Specifies the provisioning type for the catalog. Values can be:

- o Manual - No provisioning.
- o PVS - Machine provisioned by PVS (machine may be physical, blade, VM,...).
- o MCS - Machine provisioned by MCS (machine must be VM).

Required?	false
Default Value	
Accept Pipeline Input?	false

-PublishedApplication<String>

Gets machines with a specific application published to them (identified by its browser name).

Required?	false
Default Value	
Accept Pipeline Input?	false

-PublishedName<String>

Gets desktops with a specific published name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PvdEstimatedCompletionTime<DateTime>

If preparation of the Personal vDisk is currently in progress for this machine, this reports an estimation of the time at which the process will be complete.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PvdPercentDone<Int32>

Gets machines a specific percentage through the Personal vDisk preparation process.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PvdStage<PvdStage>

Gets machines at a specific personal vDisk stage.

Valid values are None, Requested, Starting, Working and Failed.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PvdUpdateStartTime<DateTime>

If preparation of the Personal vDisk is currently in progress for this machine, this reports when the update process began.

Required?	false
Default Value	
Accept Pipeline Input?	false

-RegistrationState<RegistrationState>

Gets machines in a specific registration state.

Valid values are Unregistered, Initializing, Registered, and AgentError.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ScheduledReboot<ScheduledReboot>

Gets machines according to their current status with respect to any scheduled reboots (for either scheduled desktop group reboots or image rollout purposes). Valid values are:

- o None - No reboot currently scheduled.
- o Pending - Reboot scheduled but machine still available for use.
- o Draining - Reboot scheduled. New logons are disabled, but reconnections to existing sessions are allowed.
- o InProgress - Machine is actively being rebooted.
- o Natural - Natural reboot in progress. Machine is awaiting a restart.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SecureIcaRequired<Boolean>

Gets machines configured with a particular SecureIcaRequired setting. Note that the machine setting of \$null indicates that the desktop group value is used.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionAutonomouslyBrokered<Boolean>

Gets machines according to whether their current session is autonomously brokered or not. Autonomously brokered sessions are HDX sessions established by direct connection without being brokered.

Session properties are always null for multi-session machines.

Required?	false

Default Value	
Accept Pipeline Input?	false

-SessionClientAddress<String>

Gets machines with a specific client IP address.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionClientName<String>

Gets machines with a specific client name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionClientVersion<String>

Gets machines with a specific client version.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionConnectedViaHostName<String>

Gets machines with a specific host name of the incoming connection. This is usually a proxy or Citrix Access Gateway server.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionConnectedViaIP<String>

Gets machines with a specific IP address of the incoming connection.

Session properties are always null for multi-session machines.

Required?	false

Default Value	
Accept Pipeline Input?	false

-SessionCount<Int32>

Gets machines according to the total number of both pending and established user sessions on the machine.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionDeviceId<String>

Gets machines with a specific client device ID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionHardwareId<String>

Gets machines with a specific client hardware ID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionHidden<Boolean>

Gets machines by whether their sessions are hidden or not. Hidden sessions are treated as though they do not exist when launching sessions using XenDesktop; a hidden session cannot be reconnected to, but a new session may be launched using the same entitlement.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionKey<Guid>

Gets machine running the session with the specified unique key.

Session properties are always null for multi-session machines.

Required?	false
Default Value	

Accept Pipeline Input?	false

-SessionLaunchedViaHostName<String>

Gets machines with a specific host name of the Web Interface server from which the user launched the session.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionLaunchedViaIP<String>

Gets machines with a specific IP address of the Web Interface server from which the user launched the session.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionProtocol<String>

Gets machines with connections using a specific protocol, for example HDX, RDP, or Console.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionSecureIcaActive<Boolean>

Gets machines depending on whether the current session uses SecureICA or not.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionsEstablished<Int32>

Gets machines according to the number of established user sessions present on the machine.

Required?	false

Default Value	
Accept Pipeline Input?	false

-SessionSmartAccessTag<String>

Gets session machines where the session has the specific SmartAccess tag.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionsPending<Int32>

Get machines according to the number of pending user sessions for the machine.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionStartTime<DateTime>

Gets machines with a specific session start time.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionState<SessionState>

Gets machines with a specific session state.

Valid values are \$null, Other, PreparingSession, Connected, Active, Disconnected, Reconnecting, NonBrokeredSession, and Unknown.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionStateChangeTime<DateTime>

Gets machines whose sessions last changed state at a specific time.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionSupport<SessionSupport>

Gets machines that have the specified session capability. Values can be:

- o SingleSession - Single-session only machine.
- o MultiSession - Multi-session capable machine.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionType<SessionType>

Gets machines with a specific session state.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionUid<Int64>

Gets single-session machines with a specific session UID (\$null for no session).

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionUserName<String>

Gets machines with a specific user name for the current session (in the form 'domain\user').

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionUserSID<String>

Gets machines with a specific SID of the current session user.

Session properties are always null for multi-session machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SID<String>

Gets machines with a specific machine SID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SummaryState<DesktopSummaryState>

Gets machines with a specific summary state.

Valid values are Off, Unregistered, Available, Disconnected, and InUse.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SupportedPowerActions<String[]>

A list of power actions supported by this machine.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Tag<String>

Gets machines where the session has the given SmartAccess tag.

Required?	false
Default Value	
Accept Pipeline Input?	false

-UUID<Guid>

Gets machines with the specified value of UUID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-VMToolsState<VMToolsState>

Gets machines with a specific VM tools state.

Valid values are NotPresent, Unknown, NotStarted, and Running.

Required?	false
Default Value	
Accept Pipeline Input?	false

-WillShutdownAfterUse<Boolean>

Gets machines depending on whether they shut down after use or not.

Required?	false
Default Value	
Accept Pipeline Input?	false

-WindowsConnectionSetting<WindowsConnectionSetting>

Gets machines according to their current Windows connection setting (logon mode). Valid values are:

- o LogonEnabled - All logons are enabled.
- o Draining - New logons are disabled, but reconnections to existing sessions are allowed.
- o DrainingUntilRestart - Same as Draining, but setting reverts to LogonEnabled when machine next restarts.
- o LogonDisabled - All logons and reconnections are disabled.

This is a Windows setting and is not controlled by XenDesktop. It applies only to multi-session machines; for single-session machines its value is always LogonEnabled.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AssignedUserSID<String>

Gets machines with the specific SID of the user to whom the desktop is assigned.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about_Broker_Filtering](#) for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by **-ReturnTotalRecordCount**.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-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.
```

Group-BrokerSession

May 04, 2016

Groups and counts sessions with the same value for a specified property.

Syntax

```
Group-BrokerSession [-Uid] <Int64> -Property <String> [-AdminAddress <String>] [<CommonParameters>]
```

```
Group-BrokerSession -Property <String> [-SessionKey <Guid>] [-AgentVersion <String>] [-ApplicationInUse <String>] [-AppState <SessionAppState>] [-AppStateLastChangeTime <DateTime>] [-AutonomouslyBrokered <Boolean>] [-BrokeringDuration <Int32>] [-BrokeringTime <DateTime>] [-BrokeringUserName <String>] [-BrokeringUserSID <String>] [-CatalogName <String>] [-ClientAddress <String>] [-ClientName <String>] [-ClientPlatform <String>] [-ClientProductId <Int32>] [-ClientVersion <String>] [-ConnectedViaHostName <String>] [-ConnectedViaIP <String>] [-ConnectionMode <ConnectionMode>] [-ControllerDNSName <String>] [-DesktopGroupName <String>] [-DesktopGroupUid <Int32>] [-DesktopKind <DesktopKind>] [-DesktopSID <String>] [-DesktopUid <Int32>] [-DeviceId <String>] [-DNSName <String>] [-EstablishmentDuration <Int32>] [-EstablishmentTime <DateTime>] [-HardwareId <String>] [-Hidden <Boolean>] [-HostedMachineName <String>] [-HostingServerName <String>] [-HypervisorConnectionName <String>] [-ImageOutOfDate <Boolean>] [-InMaintenanceMode <Boolean>] [-IPAddress <String>] [-IsAnonymousUser <Boolean>] [-IsPhysical <Boolean>] [-LaunchedViaHostName <String>] [-LaunchedViaIP <String>] [-LogoffInProgress <Boolean>] [-LogonInProgress <Boolean>] [-MachineName <String>] [-MachineSummaryState <DesktopSummaryState>] [-MachineUid <Int32>] [-Metadata <String>] [-OSType <String>] [-PersistUserChanges <PersistUserChanges>] [-PowerState <PowerState>] [-Protocol <String>] [-ProvisioningType <ProvisioningType>] [-ReceiverIPAddress <String>] [-ReceiverName <String>] [-SecureIcaActive <Boolean>] [-SessionId <Int32>] [-SessionState <SessionState>] [-SessionStateChangeTime <DateTime>] [-SessionSupport <SessionSupport>] [-SessionType <SessionType>] [-StartTime <DateTime>] [-UntrustedUserName <String>] [-UserFullName <String>] [-UserName <String>] [-UserSID <String>] [-UserUPN <String>] [-ApplicationUid <Int32>] [-SharedDesktopUid <Int32>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Filters sessions using the specified criteria, then groups and counts matching sessions with the same value for a particular property. The number of sessions in the group, and the property value for the group, is output. For example:

```
C:\PS> Group-BrokerSession -Property SessionState
```

```
Count Name
```

```
-----
```

```
43 Active
```

```
17 NonBrokeredSession
```

```
3 Disconnected
```

Filtering supports the same options as the Get-BrokerSession cmdlet, and allows filtering on both machine and session properties.

Group-BrokerSession is similar to the standard PowerShell Group-Object, but is faster than piping the output of Get-

BrokerSession into Group-Object when working with many machines.

Note that the MaxRecordCount, ReturnTotalRecordCount, Skip, and SortBy parameters apply to GroupInfo records output rather than the filtered sessions.

Related topics

[Get-BrokerSession](#)

Group-Object

Parameters

-Uid<Int64>

Get session by its Uid.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Property<String>

Selects the property by which matching sessions are grouped.

Required?	true
Default Value	
Accept Pipeline Input?	false

-SessionKey<Guid>

Gets session having the specified unique key.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AgentVersion<String>

Gets sessions with a specific Virtual Desktop Agent version.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ApplicationInUse<String>

Gets sessions running specific applications (identified by their SDK Name property).

Required?	false
Default Value	
Accept Pipeline Input?	false

-AppState<SessionAppState>

Get sessions by their app state.

Valid values are PreLogon, PreLaunched, Active, Desktop, Lingering and NoApps.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AppStateLastChangeTime<DateTime>

Get sessions by their app state change time.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AutonomouslyBrokered<Boolean>

Gets sessions according to whether they are autonomously brokered or not. Autonomously brokered sessions are HDX sessions established by direct connection without being brokered.

Required?	false
Default Value	
Accept Pipeline Input?	false

-BrokeringDuration<Int32>

Gets session with a specific time taken to broker. In general, Citrix recommends using -Filter and relative comparisons.

Required?	false
Default Value	
Accept Pipeline Input?	false

-BrokeringTime<DateTime>

Get sessions brokered at a specific time. In general, Citrix recommends using -Filter and relative comparisons.

Required?	false
Default Value	
Accept Pipeline Input?	false

-BrokeringUserName<String>

Get sessions by brokering user.

Required?	false
Default Value	
Accept Pipeline Input?	false

-BrokeringUserSID<String>

Get sessions by brokering user SID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-CatalogName<String>

Gets sessions on machines from a specific catalog name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ClientAddress<String>

Get sessions by client IP address.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Client Name<String>

Get sessions by client name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ClientPlatform<String>

Get sessions by client platform.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Client Product Id<Int32>

Get sessions by client product ID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Client Version<String>

Get sessions by client version.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ConnectedViaHostName<String>

Get sessions by host name of the incoming connection. This is usually a proxy or Citrix Access Gateway server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ConnectedViaIP<String>

Get sessions by IP address of the incoming connection.

Required?	false

Default Value	
Accept Pipeline Input?	false

-ConnectionMode<ConnectionMode>

Gets sessions by the way in which the most recent connection to the session was established.

Valid modes are Brokered, Unbrokered, LeasedConnection, VdaHighAvailabilityMode, ThirdPartyBroker, and ThirdPartyBrokerWithLicensing.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ControllerDNSName<String>

Gets sessions that are hosted on machines which are registered with a specific controller.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopGroupName<String>

Gets sessions from a desktop group with the specified name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopGroupUid<Int32>

Gets sessions from a desktop group with the specified UID.

Required?	false

Default Value	
Accept Pipeline Input?	false

-DesktopKind<DesktopKind>

Gets sessions on a desktop of a particular kind.

Valid values are Private and Shared.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopSID<String>

Get sessions by desktop SID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopUid<Int32>

Get sessions by desktop Uid.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DeviceId<String>

Get sessions by client device id.

Required?	false

Default Value	
Accept Pipeline Input?	false

-DNSName<String>

Gets sessions by their machine's DNS name.

Required?	false
Default Value	
Accept Pipeline Input?	false

-EstablishmentDuration<Int32>

Gets sessions which took a specific time to establish. In general, Citrix recommends using -Filter and relative comparisons.

Required?	false
Default Value	
Accept Pipeline Input?	false

-EstablishmentTime<DateTime>

Gets sessions which became established at a particular time. In general, Citrix recommends using -Filter and relative comparisons.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HardwareId<String>

Get sessions by client hardware id.

Required?	false

Default Value	
Accept Pipeline Input?	false

-Hidden<Boolean>

Get sessions by whether they are hidden or not. Hidden sessions are treated as though they do not exist when brokering sessions; a hidden session cannot be reconnected to, but a new session may be launched using the same entitlement.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HostedMachineName<String>

Gets sessions by their machine's name as known to its hypervisor.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HostingServerName<String>

Gets sessions hosted by a machine with a specific name of the hosting hypervisor server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HypervisorConnectionName<String>

Gets sessions hosted by a machine with a specific name of the hosting hypervisor connection.

Required?	false

Default Value	
Accept Pipeline Input?	false

-ImageOutOfDate<Boolean>

Gets sessions hosted by a machine with a specific ImageOutOfDate setting.

Required?	false
Default Value	
Accept Pipeline Input?	false

-InMaintenanceMode<Boolean>

Gets sessions hosted by a machine with a specific InMaintenanceMode setting.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IPAddress<String>

Gets sessions hosted by a machine with a specific IP address.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IsAnonymousUser<Boolean>

Gets sessions depending on whether they were established anonymously (\$true) or not (\$false). An anonymous session is established without user credentials and a temporary local user account is used.

Required?	false
Default Value	

Accept Pipeline Input?	false

-IsPhysical<Boolean>

Gets sessions hosted on machines where the flag indicating if the machine can be power managed by the Citrix Broker Service matches the requested value. Where the power state of the machine cannot be controlled, specify \$true, otherwise \$false.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LaunchedViaHostName<String>

Get sessions by the host name of the Web Interface server from which a user launches a session.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LaunchedViaIP<String>

Get sessions by the IP address of the Web Interface server from which a user launches a session.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LogoffInProgress<Boolean>

Gets sessions by whether they are in the process of being logged off or not.

Required?	false
Default Value	

Accept Pipeline Input?	false

-LogonInProgress<Boolean>

Gets sessions by whether they are still executing user logon processing or not.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MachineName<String>

Gets sessions by their machine name (in the form DOMAIN\machine).

Required?	false
Default Value	
Accept Pipeline Input?	false

-MachineSummaryState<DesktopSummaryState>

Gets sessions on a machine with a specific summary state.

Valid values are Off, Unregistered, Available, Disconnected, Preparing, and InUse.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MachineUid<Int32>

Gets sessions on a machine with the specified UID.

Required?	false
Default Value	

Accept Pipeline Input?	false

-Metadata<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

Required?	false
Default Value	
Accept Pipeline Input?	false

-OSType<String>

Gets sessions with a specific type of operating system.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PersistUserChanges<PersistUserChanges>

Gets sessions where the user changes are persisted in a particular manner. Values can be:

- o OnLocal - User changes are persisted locally.
- o Discard - User changes are discarded.
- o OnPvd - User changes are persisted on the Pvd.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PowerState<PowerState>

Gets sessions on machines in the specified power state.

Valid values are Unmanaged, Unknown, Unavailable, On, Suspended, TurningOn, TurningOff, Suspending, and Resuming.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Protocol<String>

Get sessions by connection protocol. Valid values are HDX, RDP and Console.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ProvisioningType<ProvisioningType>

Gets sessions hosted on machines provisioned in a particular manner. Values can be:

- o Manual - No automated provisioning.
- o PVS - Machine provisioned by PVS (machine may be physical, blade, VM,...).
- o MCS - Machine provisioned by MCS (machine must be VM).

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReceiverIPAddress<String>

Gets sessions with the specified client IP address supplied by Receiver (for example, StoreFront) when the session was launched, or reconnected.

Required?	false
Default Value	

Accept Pipeline Input?	false

-ReceiverName<String>

Gets sessions with the specified client name supplied by Receiver (for example, StoreFront) when the session was launched, or reconnected.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SecureIcaActive<Boolean>

Get sessions by their use of SecureICA.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionId<Int32>

Deprecated.

Gets sessions by session ID, a unique identifier that Remote Desktop Services uses to track the session but it is only unique on that machine.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionState<SessionState>

Get sessions by their state.

Valid values are Other, PreparingNewSession, Connected, Active, Disconnected, Reconnecting, NonBrokeredSession, and Unknown.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionStateChangeTime<DateTime>

Get sessions by their last state change time. In general, Citrix recommends using -Filter and relative comparisons.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionSupport<SessionSupport>

Gets sessions hosted on machines which support the required pattern of sessions. Values can be:

- o SingleSession - Single-session only machine.
- o MultiSession - Multi-session capable machine.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionType<SessionType>

Get sessions by their type.

Valid values are Application and Desktop.

Required?	false
Default Value	
Accept Pipeline Input?	false

-StartTime<DateTime>

Get sessions by their start time. In general, Citrix recommends using -Filter and relative comparisons.

Required?	false
Default Value	
Accept Pipeline Input?	false

-UntrustedUserName<String>

Gets sessions by the untrusted user name reported directly from the machine (in the form DOMAIN\user).

Required?	false
Default Value	
Accept Pipeline Input?	false

-UserFullName<String>

Gets sessions by user's full name (usually 'first-name last-name').

Required?	false
Default Value	
Accept Pipeline Input?	false

-UserName<String>

Get sessions by user name (in the form DOMAIN\user).

Required?	false
Default Value	
Accept Pipeline Input?	false

-UserSID<String>

Get sessions by user's Windows SID.

Required?	false
Default Value	
Accept Pipeline Input?	false

-UserUPN<String>

Gets sessions by user's User Principal Name (in the form user@domain).

Required?	false
Default Value	
Accept Pipeline Input?	false

-ApplicationUid<Int32>

Get sessions running the application with the specified Uid.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SharedDesktopUid<Int32>

Get sessions by SharedDesktop Uid.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error

record is additional information and does not affect the objects written to the output pipeline. See about_Broker_Filtering for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell style filter expression. See `about_Broker_Filtering` for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.GroupInfo

Each GroupInfo object represents one group, and contains the following properties:

-- Count: The count of sessions in this group.

-- Name: The value of the property the sessions were grouped by (as a string).

If you do not specify `-SortBy`, groups are sorted with the largest count first.

Notes

To compare dates or times, use `-Filter` and relative comparisons. For more information, see `about_Broker_Filtering` and the examples.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Group-BrokerSession -Property SessionState -DesktopGroupName dg1
```

Group sessions on machines from the dg1 group by session state.

----- **EXAMPLE 2** -----

```
C:\PS> Group-BrokerSession -Property ClientName -ClientName 'ThinClient*' -SortBy Name
```

List alphabetically the names of the clients connected to the site, but only show clients whose names starts with 'ThinClient'.

Import-BrokerDesktopPolicy

May 04, 2016

Sets the site wide Citrix Group Policy settings for the site.

Syntax

```
Import-BrokerDesktopPolicy [-Policy] <Byte[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Import-BrokerDesktopPolicy sets the site wide Citrix Group Policy settings. A successful call to this cmdlet will result in the supplied data being uploaded to every machine in the site prior to its next session launch.

Related topics

[Export-BrokerDesktopPolicy](#)

[New-BrokerConfigurationSlot](#)

[New-BrokerMachineConfiguration](#)

Parameters

-Policy<Byte[]>

The configuration data containing the Citrix Group Policy settings to apply to every machine in the site.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

System.Byte[] The configuration data as an opaque binary blob.

Return Values

None

Notes

Import-BrokerDesktopPolicy performs a specialized operation. Direct usage of it in scripts is discouraged, and could result in data corruption. It is recommended that this operation be performed via the Citrix Studio.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Import-BrokerDesktopPolicy $policyData
```

This command sets the Citrix Group Policy settings in the site. These policy settings are then applied to every machine prior to the next session launch.

Move-BrokerAdminFolder

May 04, 2016

Moves a folder to another place in the hierarchy, optionally renaming it

Syntax

```
Move-BrokerAdminFolder [-InputObject] <AdminFolder[]> [-Destination] <AdminFolder> [-NewName <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Move-BrokerAdminFolder [-Name] <String> [-Destination] <AdminFolder> [-NewName <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Move-BrokerAdminFolder cmdlet moves a folder for organising objects for administration purposes (for example, Applications) to another position in the hierarchy.

The following special characters are not allowed in the new FolderName: \ / ; : # . * ? = < > | [] () " ' `

Related topics

[Get-BrokerAdminFolder](#)

[New-BrokerAdminFolder](#)

Parameters

-InputObject<AdminFolder[]>

The folder(s) to be moved

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

A pattern matching the names of folders to be moved

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Destination<AdminFolder>

The destination folder the folder being moved should end up in

Required?	true
Default Value	
Accept Pipeline Input?	false

-NewName<String>

The name the new folder should have in the destination folder

Required?	false
Default Value	
Accept Pipeline Input?	false

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Depends on parameter Parameters can be piped by property name.

Return Values

None or Citrix.Broker.Admin.SDK.AdminFolder

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.AdminFolder object.

Examples

----- **EXAMPLE 1** -----

```
Move-BrokerAdminFolder F1\XXX\ F2\
```

Moves the folder called XXX within the folder F1\ to a new home in F2\

----- **EXAMPLE 2** -----

```
Move-BrokerAdminFolder F1\XXX\ F2\ -NewName YYY
```

Moves the folder called XXX within the folder F1\ to a new home in F2\ renaming it to YYY in the process

Move-BrokerApplication

May 04, 2016

Move a published application from one admin folder to another

Syntax

```
Move-BrokerApplication [-InputObject] <Application[]> [-Destination] <AdminFolder> [-NewName <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Move-BrokerApplication [-Name] <String> [-Destination] <AdminFolder> [-NewName <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Move-BrokerApplication cmdlet moves a published application from one place to another in the tree of admin folders, optionally renaming it in the process (if you only want to change the name of the application for administrative purposes and not its location in the tree, use the Rename-BrokerApplication cmdlet).

The location and name of a published application in this sense is only of interest to the administrator, changes do not affect the end-user experience.

Related topics

[New-BrokerApplication](#)

[Add-BrokerApplication](#)

[Get-BrokerApplication](#)

[Remove-BrokerApplication](#)

[Rename-BrokerApplication](#)

[Set-BrokerApplication](#)

Parameters

-InputObject<Application[]>

The application(s) to be moved

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

The application(s) to be moved

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Destination<AdminFolder>

The destination location within the admin folder hierarchy

Required?	true
Default Value	
Accept Pipeline Input?	false

-NewName<String>

The new name of the application in its new destination

Required?	false
Default Value	
Accept Pipeline Input?	false

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.Application You can pipe applications to Move-BrokerApplication.

Return Values

None or Citrix.Broker.Admin.SDK.Application

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.Application object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Move-BrokerApplication -Name 'App1' -Destination 'F1\'
Moves the application in the root folder called "App1" to the folder "F1\".
```

----- EXAMPLE 2 -----

```
C:\PS> Move-BrokerApplication 'F1\App1' 'F2\' -NewName 'Application1'
Moves the application in folder "F1" called "App1" to the folder "F2", renaming it to "Application1" in the process.
```


New-BrokerAccessPolicyRule

May 04, 2016
Creates a new rule in the site's access policy.

Syntax

New-BrokerAccessPolicyRule [-Name] <String> -DesktopGroupUid <Int32> [-AllowedConnections <AllowedConnection>] [-AllowedProtocols <String[]>] [-AllowedUsers <AllowedUser>] [-AllowRestart <Boolean>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedClientIPFilterEnabled <Boolean>] [-ExcludedClientIPs <IPAddressRange[]>] [-ExcludedClientNameFilterEnabled <Boolean>] [-ExcludedClientNames <String[]>] [-ExcludedSmartAccessFilterEnabled <Boolean>] [-ExcludedSmartAccessTags <String[]>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-HdxSslEnabled <Boolean>] [-IncludedClientIPFilterEnabled <Boolean>] [-IncludedClientIPs <IPAddressRange[]>] [-IncludedClientNameFilterEnabled <Boolean>] [-IncludedClientNames <String[]>] [-IncludedSmartAccessFilterEnabled <Boolean>] [-IncludedSmartAccessTags <String[]>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

New-BrokerAccessPolicyRule [-Name] <String> -IncludedDesktopGroups <DesktopGroup[]> [-IncludedDesktopGroupFilterEnabled <Boolean>] [-AllowedConnections <AllowedConnection>] [-AllowedProtocols <String[]>] [-AllowedUsers <AllowedUser>] [-AllowRestart <Boolean>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedClientIPFilterEnabled <Boolean>] [-ExcludedClientIPs <IPAddressRange[]>] [-ExcludedClientNameFilterEnabled <Boolean>] [-ExcludedClientNames <String[]>] [-ExcludedSmartAccessFilterEnabled <Boolean>] [-ExcludedSmartAccessTags <String[]>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-HdxSslEnabled <Boolean>] [-IncludedClientIPFilterEnabled <Boolean>] [-IncludedClientIPs <IPAddressRange[]>] [-IncludedClientNameFilterEnabled <Boolean>] [-IncludedClientNames <String[]>] [-IncludedSmartAccessFilterEnabled <Boolean>] [-IncludedSmartAccessTags <String[]>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Detailed Description

The New-BrokerAccessPolicyRule cmdlet adds a new rule to the site's access policy.

An access policy rule defines a set of connection filters and access control rights relating to a desktop group. These allow fine-grained control of what access is granted to a desktop group based on details of, for example, a user's endpoint device, its address, and the user's identity.

Multiple rules in the access policy can apply to the same desktop group.

For a user to gain access to a desktop group via a rule their connection must match all its enabled include filters, and none of its enabled exclude filters. In addition, for a user to be able to launch a desktop or application resource session from the desktop group, they must have an entitlement to use the resource granted by the entitlement or assignment policies, or by direct machine assignment.

Related topics

- [Get-BrokerAccessPolicyRule](#)
- [Set-BrokerAccessPolicyRule](#)
- [Rename-BrokerAccessPolicyRule](#)
- [Remove-BrokerAccessPolicyRule](#)

Parameters

-Name<String>

Specifies the administrative name of the new rule. Each rule within the site's access policy must have a unique name.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-DesktopGroupUid<Int32>

Specifies the desktop group to which the new rule applies.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-IncludedDesktopGroups<DesktopGroup[]>

This parameter is supported for backward compatibility only. If used only a single desktop group UID can be specified.

The IncludedDesktopGroups and IncludedDesktopGroupFilterEnabled parameters have been superseded by the DesktopGroupUid parameter.

Required?	true
Default Value	(empty list)
Accept Pipeline Input?	true (ByPropertyName)

-AllowedConnections<AllowedConnection>

Specifies whether connections must be local or via Access Gateway, and if so whether specified SmartAccess tags must be provided by Access Gateway with the connection. This property forms part of the included SmartAccess tags filter.

Valid values are Filtered, NotViaAG, and ViaAG.

For a detailed description of this property see "help about_Broker_AccessPolicy".

Required?	false
Default Value	Filtered
Accept Pipeline Input?	true (ByPropertyName)

-AllowedProtocols<String[]>

Specifies the protocols (for example HDX, RDP) available to the user for sessions delivered from the new rule's desktop group. If the user gains access to a desktop group by multiple rules, the allowed protocol list is the combination of the protocol lists from all those rules.

If the protocol list is empty, access to the desktop group is implicitly denied.

Required?	false
Default Value	HDX
Accept Pipeline Input?	true (ByPropertyName)

-AllowedUsers<AllowedUser>

Specifies the behavior of the included users filter of the new rule. This can restrict access to a list of named users or groups, allow access to any authenticated user, any user (whether authenticated or not), or only non-authenticated users. For a detailed description of this property see "help about_Broker_AccessPolicy".

Valid values are Filtered, AnyAuthenticated, Any, AnonymousOnly and FilteredOrAnonymous.

Required?	false
Default Value	Filtered
Accept Pipeline Input?	true (ByPropertyName)

-AllowRestart<Boolean>

Specifies if the user can restart sessions delivered from the new rule's desktop group. Session restart is handled as follows: For sessions on single-session power-managed machines, the machine is powered off, and a new session launch request made; for sessions on multi-session machines, a logoff request is issued to the session, and a new session launch request made; otherwise the property is ignored.

Required?	false
Default Value	false

Accept Pipeline Input?	true (ByPropertyName)

-Description<String>

Specifies an optional description of the new rule. The text is purely informational for the administrator, it is never visible to the end user.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Enabled<Boolean>

Specifies whether the new rule is initially enabled. A disabled rule is ignored when evaluating the site's access policy.

Required?	false
Default Value	true
Accept Pipeline Input?	true (ByPropertyName)

-ExcludedClientIPFilterEnabled<Boolean>

Specifies whether the excluded client IP address filter is initially enabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-ExcludedClientIPs<IPAddressRange[]>

Specifies IP addresses of user devices explicitly denied access to the new rule's desktop group. Addresses can be specified as simple numeric addresses or as subnet masks (for example, 10.40.37.5 or 10.40.0.0/16). This property forms part of the excluded client IP address filter.

Required?	false
Default Value	(empty list)
Accept Pipeline Input?	true (ByPropertyName)

-ExcludedClientNameFilterEnabled<Boolean>

Specifies whether the excluded client names filter is initially enabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-ExcludedClientNames<String[]>

Specifies names of user devices explicitly denied access to the new rule's desktop group. This property forms part of the excluded client names filter.

Required?	false
Default Value	(empty list)
Accept Pipeline Input?	true (ByPropertyName)

-ExcludedSmartAccessFilterEnabled<Boolean>

Specifies whether the excluded SmartAccess tags filter is initially enabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-ExcludedSmartAccessTags<String[]>

Specifies SmartAccess tags which explicitly deny access to the new rule's desktop group if any occur in those provided by Access Gateway with the user's connection. This property forms part of the excluded SmartAccess tags filter.

Required?	false
Default Value	(empty list)
Accept Pipeline Input?	true (ByPropertyName)

-ExcludedUserFilterEnabled<Boolean>

Specifies whether the excluded users filter is initially enabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-ExcludedUsers<User[]>

Specifies any users and groups who are explicitly denied access to the new rule's desktop group. This property forms part of the excluded users filter.

Required?	false
Default Value	(empty list)
Accept Pipeline Input?	true (ByPropertyName)

-HdxSslEnabled<Boolean>

Indicates whether SSL encryption is enabled for sessions delivered from the rule's desktop group.

Required?	false
Default Value	\$false
Accept Pipeline Input?	true (ByPropertyName)

-IncludedClientIPFilterEnabled<Boolean>

Specifies whether the included client IP address filter is initially enabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-IncludedClientIPs<IPAddressRange[]>

Specifies IP addresses of user devices allowed access to the new rule's desktop group. Addresses can be specified as simple numeric addresses or as subnet masks (for example, 10.40.37.5 or 10.40.0.0/16). This property forms part of the included client IP address filter.

Required?	false
Default Value	(empty list)
Accept Pipeline Input?	true (ByPropertyName)

-IncludedClientNameFilterEnabled<Boolean>

Specifies whether the included client name filter is initially enabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-IncludedClientNames<String[]>

Specifies names of user devices allowed access to the new rule's desktop group. This property forms part of the included client names filter.

Required?	false
Default Value	(empty list)
Accept Pipeline Input?	true (ByPropertyName)

-IncludedSmartAccessFilterEnabled<Boolean>

Specifies whether the included SmartAccess tags filter is initially enabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-IncludedSmartAccessTags<String[]>

Specifies SmartAccess tags which grant access to the new rule's desktop group if any occur in those provided by Access Gateway with the user's connection. This property forms part of the excluded SmartAccess tags filter.

Required?	false
Default Value	(empty list)

Accept Pipeline Input?	true (ByPropertyName)

-IncludedUserFilterEnabled<Boolean>

Specifies whether the included users filter is initially enabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-IncludedUsers<User[]>

Specifies users and groups who are granted access to the new rule's desktop group. This property forms part of the included users filter.

Required?	false
Default Value	(empty list)
Accept Pipeline Input?	true (ByPropertyName)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

-IncludedDesktopGroupFilterEnabled<Boolean>

This parameter is supported for backward compatibility only. If used the supplied value must be \$true.

The IncludedDesktopGroups and IncludedDesktopGroupFilterEnabled parameters have been superseded by the DesktopGroupUid parameter.

Required?	false
Default Value	true
Accept Pipeline Input?	true (ByPropertyName)

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.AccessPolicyRule

New-BrokerAccessPolicyRule returns the newly created access policy rule.

Examples

----- EXAMPLE 1 -----

```
C:\PS> $dg = Get-BrokerDesktopGroup 'Tech Support'
```

```
C:\PS> New-BrokerAccessPolicyRule 'UK Tech Support' -IncludedUserFilterEnabled $true -IncludedUsers support\uk-staff -DesktopGroupUid $dg.Uid -AllowedProtocols 'HDX'
```

Creates an access policy rule allowing access to the Tech Support desktop group for all users of the SUPPORT\uk-staff group. Connections to desktop or application resources in the group can only be made using the HDX protocol.

For users to gain access to resources in the group also requires that, depending on the desktop kind of the group, appropriate assignment or entitlement policy rules, or explicit machine assignments exist.

New-BrokerAdminFolder

May 04, 2016

Creates a new admin folder.

Syntax

```
New-BrokerAdminFolder [-FolderName] <String> [-ParentFolder <AdminFolder>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The New-BrokerAdminFolder cmdlet creates a new folder for organising objects for administration purposes (for example, Applications).

New-BrokerAdminFolder creates the folder object and optionally places it within an existing admin folder if required.

The following special characters are not allowed in the FolderName: \ / ; : # . * ? = < > | [] () " ' `

Related topics

[Get-BrokerAdminFolder](#)

[Remove-BrokerAdminFolder](#)

Parameters

-FolderName<String>

The simple name of the new folder within its parent (if any)

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-ParentFolder<AdminFolder>

The name or UID of the parent folder (if any)

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Depends on parameter Parameters can be piped by property name.

Return Values

Citrix.Broker.Admin.SDK.AdminFolder

The new admin folder.

Examples

----- EXAMPLE 1 -----

New-BrokerAdminFolder F1

Creates an admin folder called F1 under the root folder (i.e. F1\)

----- EXAMPLE 2 -----

New-BrokerAdminFolder F2 -AdminFolder F1\

Creates an admin folder called F2 under the folder F1\ (i.e. F1\F2\)

New-BrokerAppAssignmentPolicyRule

May 04, 2016

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

May 04, 2016

Creates a new application rule in the site's entitlement policy.

Syntax

```
New-BrokerAppEntitlementPolicyRule [-Name] <String> -DesktopGroupUid <Int32> [-Description <String>] [-Enabled <Boolean>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-SessionReconnection <SessionReconnection>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The New-BrokerAppEntitlementPolicyRule cmdlet adds a new application rule to the site's entitlement policy.

An application rule in the entitlement policy defines the users who are allowed per-session access to a machine to run one or more applications published from the rule's desktop group.

The following constraints apply when creating an application entitlement rule for a desktop group:

- o The group's desktop kind must be Shared
- o The group's delivery type must be AppsOnly or DesktopsAndApps
- o Only a single application rule can apply to a given group

When a user selects an application published from a shared group, a machine is selected from the group on which to run the application. No permanent association exists between the user and the selected machine; once the session ends the association also ends.

Even though only a single application entitlement and therefore session can be defined for a group, the user can still run multiple applications from the group because the applications run within the same session.

Related topics

[Get-BrokerAppEntitlementPolicyRule](#)

[Set-BrokerAppEntitlementPolicyRule](#)

[Rename-BrokerAppEntitlementPolicyRule](#)

[Remove-BrokerAppEntitlementPolicyRule](#)

Parameters

-Name<String>

Specifies the administrative name of the new application rule. Each rule in the site's entitlement policy must have a unique name (irrespective of whether they are desktop or application rules).

Required?	true

Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-DesktopGroupUid<Int32>

Specifies the unique ID of the desktop group to which the new application rule applies.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Description<String>

Specifies an optional description of the new application rule. The text is purely informational for the administrator, it is never visible to the end user.

Required?	false
Default Value	null
Accept Pipeline Input?	true (ByPropertyName)

-Enabled<Boolean>

Specifies whether the new application rule is initially enabled. A disabled rule is ignored when evaluating the site's entitlement policy.

Required?	false
Default Value	true
Accept Pipeline Input?	true (ByPropertyName)

-ExcludedUserFilterEnabled<Boolean>

Specifies whether the excluded users filter is initially enabled. If the filter is disabled then any user entries in the filter are ignored when entitlement policy rules are evaluated.

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-ExcludedUsers<User[]>

Specifies the excluded users filter of the application rule, that is, the users and groups who are explicitly denied entitlements to published applications from the desktop group.

This can be used to exclude users or groups who would otherwise gain access by groups specified in the included users filter.

Required?	false
Default Value	(empty list)
Accept Pipeline Input?	true (ByPropertyName)

-IncludedUserFilterEnabled<Boolean>

Specifies whether the included users filter is initially enabled. If the filter is disabled then any user who satisfies the requirements of the access policy is implicitly granted an entitlement to an application session by the new rule.

Users who would be implicitly granted access when the filter is disabled can still be explicitly denied access using the excluded users filter.

Required?	false
Default Value	true
Accept Pipeline Input?	true (ByPropertyName)

-IncludedUsers<User[]>

Specifies the included users filter of the application rule, that is, the users and groups who are granted an entitlement to an application session by the new rule.

If a user appears explicitly in the excluded users filter of the rule or is a member of a group that appears in the excluded users filter, no entitlement is granted whether or not the user appears in the included users filter.

Required?	false
Default Value	(empty list)

Accept Pipeline Input?	true (ByPropertyName)

-SessionReconnection<SessionReconnection>

Defines reconnection (roaming) behavior for sessions launched using this rule. Session reconnection control is an experimental and unsupported feature.

Required?	false
Default Value	Always
Accept Pipeline Input?	true (ByPropertyName)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.AppEntitlementPolicyRule

New-BrokerAppEntitlementPolicyRule returns the newly created application rule in the entitlement policy.

Examples

----- EXAMPLE 1 -----

```
C:\PS> $dg = Get-BrokerDesktopGroup 'Customer Support'
```

```
C:\PS> New-BrokerAppEntitlementPolicyRule 'UK Office' -DesktopGroupUid $dg.Uid -IncludedUsers support\uk-staff
```

Creates an application rule in the entitlement policy that entitles all members of the SUPPORT\uk-staff group to a machine for running applications published from the Customer Support desktop group.

New-BrokerApplication

May 04, 2016
Creates a new published application.

Syntax

New-BrokerApplication [-Name] <String> -CommandLineExecutable <String> -DesktopGroup <DesktopGroup> [-AdminFolder <AdminFolder>] [-ApplicationType <ApplicationType>] [-BrowserName <String>] [-ClientFolder <String>] [-CommandLineArguments <String>] [-CpuPriorityLevel <CpuPriorityLevel>] [-Description <String>] [-Enabled <Boolean>] [-IconFromClient <Boolean>] [-IconUid <Int32>] [-Priority <Int32>] [-PublishedName <String>] [-SecureCmdLineArgumentsEnabled <Boolean>] [-ShortcutAddedToDesktop <Boolean>] [-ShortcutAddedToStartMenu <Boolean>] [-StartMenuFolder <String>] [-UserFilterEnabled <Boolean>] [-UUID <Guid>] [-Visible <Boolean>] [-WaitForPrinterCreation <Boolean>] [-WorkingDirectory <String>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Detailed Description

The New-BrokerApplication cmdlet creates a new published application in the site.

New-BrokerApplication creates the application object, and associates it with a desktop group. Application objects have three names that identify them (in addition to their Uid): the Name, BrowserName and the PublishedName. The BrowserName is unique across the entire site, and is primarily used internally. The Name is also unique and is what is seen by the administrator; it contains any prefix for an enclosing admin folder (if any). The PublishedName is not unique and is what is seen by the users.

You can create both HostedOnDesktop and InstalledOnClient applications but the ApplicationType cannot be changed later.

The following special characters are not allowed in the Name, BrowserName or the PublishedName properties: \ / ; : # . * ? = < > | [] () " ' "

In addition the ` character is not allowed in the Name property.

See about_Broker_Applications for more information.

Related topics

- [Add-BrokerApplication](#)
- [Remove-BrokerApplication](#)
- [Get-BrokerApplication](#)
- [Remove-BrokerApplication](#)
- [Rename-BrokerApplication](#)
- [Move-BrokerApplication](#)
- [Set-BrokerApplication](#)

Parameters

-Name<String>
Specifies the name of the application (must be unique within folder).

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-CommandLineExecutable<String>
Specifies the name of the executable file to launch. The full path need not be provided if it's already in the path. Environment variables can also be used.

Required?	true
Default Value	(required)
Accept Pipeline Input?	true (ByPropertyName)

-DesktopGroup<DesktopGroup>

Specifies which desktop group this application should be associated with. The association between application and desktop groups can be added or removed using the Add-BrokerApplication and Remove-BrokerApplication cmdlets.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-AdminFolder<AdminFolder>

The folder in which the new application should reside (if any).

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-ApplicationType<ApplicationType>

Specifies the type of the application: HostedOnDesktop or InstalledOnClient.

Required?	false
Default Value	(required)
Accept Pipeline Input?	true (ByPropertyName)

-BrowserName<String>

Specifies the internal name for this application. It must be unique in the site.

Required?	false
Default Value	(same as Name)
Accept Pipeline Input?	true (ByPropertyName)

-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)

-Short cutAddedToDesktop<Boolean>

Specifies whether or not a shortcut to the application should be placed on the user device. This is valid only for the Citrix Online Plug-in.

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-ShortcutAddedToStartMenu<Boolean>

Specifies whether a shortcut to the application should be placed in the user's start menu on their user device.

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-StartMenuFolder<String>

Specifies the name of the start menu folder that holds the application shortcut (if any). This is valid only for the Citrix Online Plug-in. Subdirectories can be specified with '\' character. The following special characters are not allowed: / * ? < > | " .:

Required?	false
Default Value	null
Accept Pipeline Input?	true (ByPropertyName)

-UserFilterEnabled<Boolean>

Specifies whether the application's user filter is enabled or disabled. Where the user filter is enabled, the application is visible only to users who appear in the filter (either explicitly or by virtue of group membership).

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-UUID<Guid>

An optional GUID for this application.

Required?	false
Default Value	A new GUID is generated if none is supplied.
Accept Pipeline Input?	true (ByPropertyName)

-Visible<Boolean>

Specifies whether or not this application is visible to users. Note that it's possible for an application to be disabled and still visible.

Required?	false
Default Value	true
Accept Pipeline Input?	true (ByPropertyName)

-WaitForPrinterCreation<Boolean>

Specifies whether or not the session waits for the printers to be created before allowing the user to interact with the session. Note that this property cannot be set for applications of type InstalledOnClient.

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-WorkingDirectory<String>

Specifies which working directory the executable is launched from. Environment variables can be used.

Required?	false
Default Value	null
Accept Pipeline Input?	true (ByPropertyName)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Depends on parameter Parameters can be piped by property name.

Return Values

Citrix.Broker.Admin.SDK.Application

New-BrokerApplication returns an Application object.

Notes

Usually only the Name is specified with the New-BrokerApplication cmdlet, and the system chooses a BrowserName and PublishedName for you. By default the BrowserName is the same as the Name, if it is unique in the site. If not, then "-x" is appended to the name, where "x" is a number. For instance, if there is already an application with a BrowserName of "Notepad" and a new application is created with a Name of "Notepad", then the new application gets a BrowserName of "Notepad-1". If another "Notepad" is published, it has a BrowserName of "Notepad-2".

That said, the BrowserName can optionally be specified as well.

Examples

----- EXAMPLE 1 -----

```
C:\PS> New-BrokerApplication -ApplicationType HostedOnDesktop -Name "Notepad" -CommandLineExecutable "notepad.exe" -DesktopGroup PrivateDG1
```

Creates and returns an object for a published application called "Notepad" that launches "notepad.exe".

----- EXAMPLE 2 -----

```
C:\PS> $dg = Get-BrokerDesktopGroup "SharedDG1"
C:\PS> $app = New-BrokerApplication -ApplicationType HostedOnDesktop -Name "Notepad" -CommandLineExecutable "notepad.exe" -DesktopGroup $dg
C:\PS> $group = Get-BrokerDesktopGroup -Name "Shared desktop group"
C:\PS> Add-BrokerApplication $app -DesktopGroup $group
C:\PS> $fta = Get-BrokerImportedFTA -ExtensionName ".txt"
C:\PS> New-BrokerConfiguredFTA -ImportedFTA $fta -ApplicationUid $app.Uid
```

This is a much more complete example. It creates an application object to publish Notepad and associates it first with the "SharedDG1" desktop group.

Next it adds an additional desktop group (one that can host applications), and publishes the application to that desktop group. It then gets the ImportedFTA object for the .txt file-type extension (this assumes file-type associations have already been imported), and then configures it so that ".txt" is associated with the published application.

Note: The appropriate access policy and app assignment/entitlement rules must also be configured to allow access to the application.

New-BrokerAssignmentPolicyRule

May 04, 2016
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

May 04, 2016
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 how user changes are persisted on machines in the catalog. Possible values are:

- o OnLocal: User changes are stored on the machine's local storage.
- o Discard: User changes are discarded.
- o OnPvd: User changes are stored on the user's personal vDisk.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-PvsForVM<String[]>

Deprecated:

Identifies the provisioning scheme used by this catalog. To be specified in the format: ProvisioningSchemeGuid:ServiceGroupGuid. Applicable only to thin-cloned, single-image or personal vDisk catalogs.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Description<String>

A description for the catalog.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-IsRemotePC<Boolean>

Specifies whether this is to be a Remote PC catalog.

IsRemotePC can only be enabled when:

o SessionSupport is SingleSession

o MachinesArePhysical is true.

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-MachinesArePhysical<Boolean>

Specifies whether machines in the catalog can be power-managed by the Citrix Broker Service. Where the Citrix Broker Service cannot control the power state of themachine specify True, otherwise False. Can only be specified together with a provisioning type of Pvs or Manual, or if used with the legacy CatalogKind parameter only with Pvs or PvsPvd catalog kinds.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-MinimumFunctionalLevel<FunctionalLevel>

The minimum FunctionalLevel required for machines to register in the site.

Valid values are L5, L7, L7_6

Required?	false
Default Value	The FunctionalLevel of the current release (L7_6); by default no machines with less than the most current FunctionalLevel will be functional.
Accept Pipeline Input?	true (ByPropertyName)

-PvsAddress<String>

Specifies the URL of the Provisioning Services server. Only applicable to Provisioning Services or Provisioning Services-personal vDisk catalogs.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-PvsDomain<String>

Specifies the Active Directory domain of the Provisioning Services server. Only applicable to Provisioning Services or Provisioning Services-personal vDisk catalogs.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-RemotePCHypervisorConnectionUid<Int32>

Specifies the hypervisor connection to use for powering on remote PCs in this catalog (only allowed when IsRemotePC is true).

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Scope<String[]>

Specifies the name of the delegated administration scope to which the catalog belongs.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-UUID<Guid>

An optional GUID for this catalog.

Required?	false
Default Value	A new GUID is generated if none is supplied.
Accept Pipeline Input?	true (ByPropertyName)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

-ProvisioningSchemeId<Guid>

Specifies the identity of the MCS provisioning scheme the new catalog is associated with (can only be specified for new catalogs with a ProvisioningType of MCS).

Required?	false

Default Value	\$null
Accept Pipeline Input?	true (ByPropertyName)

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.Catalog

New-BrokerCatalog returns the created catalog.

Examples

----- EXAMPLE 1 -----

C:\PS> New-BrokerCatalog -AllocationType Static -CatalogKind Unmanaged -Description "Catalog1 Description" -Name "Catalog1 Name"
This command creates a catalog that can contain unmanaged physical or virtual machines that are permanently assigned to the user.

----- EXAMPLE 2 -----

C:\PS> New-BrokerCatalog -AllocationType Random -CatalogKind PowerManaged -Description "catalog 2 Description" -Name "Catalog2 Name"
This command creates a catalog that can contain power-managed machines that are randomly assigned to the user.

----- EXAMPLE 3 -----

C:\PS> New-BrokerCatalog -AllocationType Random -CatalogKind PVS -Description "PVS Catalog Desc" -Name "PVS Catalog Name" -PvsAddress "pvsServer@pvsDomain.com" -PvsDomain "pvsDor"
This command creates a catalog that can contain managed machines that are provisioned using Provisioning Services.

New-BrokerConfigurationSlot

May 04, 2016

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

May 04, 2016
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

May 04, 2016

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

May 04, 2016

Create a new desktop group for managing the brokering of groups of desktops.

Syntax

```
New-BrokerDesktopGroup [-Name] <String> -DesktopKind <DesktopKind> [-AutomaticPowerOnForAssigned <Boolean>] [-AutomaticPowerOnForAssignedDuringPeak <Boolean>] [-ColorDepth <ColorDepth>] [-DeliveryType <DeliveryType>] [-Description <String>] [-Enabled <Boolean>] [-IconUid <Int32>] [-InMaintenanceMode <Boolean>] [-IsRemotePC <Boolean>] [-MinimumFunctionalLevel <FunctionalLevel>] [-OffPeakBufferSizePercent <Int32>] [-OffPeakDisconnectAction <SessionChangeHostingAction>] [-OffPeakDisconnectTimeout <Int32>] [-OffPeakExtendedDisconnectAction <SessionChangeHostingAction>] [-OffPeakExtendedDisconnectTimeout <Int32>] [-OffPeakLogOffAction <SessionChangeHostingAction>] [-OffPeakLogOffTimeout <Int32>] [-PeakBufferSizePercent <Int32>] [-PeakDisconnectAction <SessionChangeHostingAction>] [-PeakDisconnectTimeout <Int32>] [-PeakExtendedDisconnectAction <SessionChangeHostingAction>] [-PeakExtendedDisconnectTimeout <Int32>] [-PeakLogOffAction <SessionChangeHostingAction>] [-PeakLogOffTimeout <Int32>] [-ProtocolPriority <String[]>] [-PublishedName <String>] [-Scope <String[]>] [-SecureIcaRequired <Boolean>] [-SessionSupport <SessionSupport>] [-SettlementPeriodBeforeAutoShutdown <TimeSpan>] [-ShutdownDesktopsAfterUse <Boolean>] [-TimeZone <String>] [-TurnOnAddedMachine <Boolean>] [-UUID <Guid>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The New-BrokerDesktopGroup cmdlet creates a new broker desktop group that can then be used to manage the brokering settings of all desktops within that desktop group. Once the desktop group has been created, you can create desktops in it by adding the appropriate broker machines to it using the Add-BrokerMachine or Add-BrokerMachinesToDesktopGroup cmdlets.

Desktop groups hold settings that apply to all desktops they contain.

For any automatic power management settings of a desktop group to take effect, the group's TimeZone property must be specified. Automatic power management operations include pool management (power time schemes), reboot schedules, session disconnect and logoff actions, and powering on assigned machines etc.

Related topics

[Get-BrokerDesktopGroup](#)

[Set-BrokerDesktopGroup](#)

[Rename-BrokerDesktopGroup](#)

[Remove-BrokerDesktopGroup](#)

[Add-BrokerMachine](#)

[Add-BrokerMachinesToDesktopGroup](#)

[Get-BrokerSite](#)

Parameters

-Name<String>

The name of the new broker desktop group.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-DesktopKind<DesktopKind>

The kind of desktops this group will hold. Valid values are Private and Shared.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-AutomaticPowerOnForAssigned<Boolean>

Specifies whether assigned desktops in the desktop group should be automatically started at the start of peak time periods. Only relevant for groups whose DesktopKind is Private.

Required?	false
Default Value	true
Accept Pipeline Input?	true (ByPropertyName)

-AutomaticPowerOnForAssignedDuringPeak<Boolean>

Specifies whether assigned desktops in the desktop group should be automatically started throughout peak time periods. Only relevant for groups whose DesktopKind is Private and which have AutomaticPowerOnForAssigned set to true.

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-ColorDepth<ColorDepth>

Specifies the color depth that the ICA session should use for desktops in this group. Valid values are FourBit, EightBit, SixteenBit, and TwentyFourBit.

Required?	false
Default Value	TwentyFourBit
Accept Pipeline Input?	true (ByPropertyName)

-DeliveryType<DeliveryType>

Specifies whether desktops, applications, or both, can be delivered from machines contained within the new desktop group. Desktop groups with a DesktopKind of Private cannot be used to deliver both desktops and applications. Defaults to DesktopsOnly if not specified.

Valid values are DesktopsOnly, AppsOnly, and DesktopsAndApps.

Required?	false
Default Value	DesktopsOnly
Accept Pipeline Input?	true (ByPropertyName)

-Description<String>

A description for this desktop group useful for administrators of the site.

Required?	false
Default Value	null
Accept Pipeline Input?	true (ByPropertyName)

-Enabled<Boolean>

Whether the desktop group should be in the enabled state; disabled desktop groups do not appear to users.

Required?	false
Default Value	true

Accept Pipeline Input?	true (ByPropertyName)

-IconUid<Int32>

The UID of the broker icon to be displayed to users for their desktop(s) in this desktop group.

Required?	false
Default Value	The Uid of the default desktop icon in this site - use the Get-BrokerSite cmdlet to find this value.
Accept Pipeline Input?	true (ByPropertyName)

-InMaintenanceMode<Boolean>

Whether the desktop should be created in maintenance mode; a desktop group in maintenance mode will not allow users to connect or reconnect to their desktops.

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-IsRemotePC<Boolean>

Specifies whether this is to be a Remote PC desktop group.

IsRemotePC can only be enabled when:

- o SessionSupport is SingleSession
- o DeliveryType is DesktopsOnly
- o DesktopKind is Private

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-MinimumFunctionalLevel<FunctionalLevel>

The minimum FunctionalLevel required for machines to work successfully in the desktop group.

Valid values are L5, L7, L7_6

Required?	false
Default Value	The FunctionalLevel of the current release (L7_6); by default no machines with less than the most current FunctionalLevel will be functional.
Accept Pipeline Input?	true (ByPropertyName)

-OffPeakBufferSizePercent<Int32>

The percentage of machines in the desktop group that should be kept available in an idle state outside peak hours.

Required?	false
Default Value	0
Accept Pipeline Input?	true (ByPropertyName)

-OffPeakDisconnectAction<SessionChangeHostingAction>

The action to be performed after a configurable period of a user session disconnecting outside peak hours. Possible values are Nothing, Suspend, or Shutdown

Required?	false
Default Value	Nothing
Accept Pipeline Input?	true (ByPropertyName)

-OffPeakDisconnectTimeout<Int32>

The number of minutes before the configured action should be performed after a user session disconnects outside peak hours.

Required?	false
Default Value	0
Accept Pipeline Input?	true (ByPropertyName)

-OffPeakExtendedDisconnectAction<SessionChangeHostingAction>

The action to be performed after a second configurable period of a user session disconnecting outside peak hours. Possible values are Nothing, Suspend, or Shutdown.

Required?	false
Default Value	Nothing
Accept Pipeline Input?	true (ByPropertyName)

-OffPeakExtendedDisconnectTimeout<Int32>

The number of minutes before the second configured action should be performed after a user session disconnects outside peak hours.

Required?	false
Default Value	0
Accept Pipeline Input?	true (ByPropertyName)

-OffPeakLogOffAction<SessionChangeHostingAction>

The action to be performed after a configurable period of a user session ending outside peak hours. Possible values are Nothing, Suspend, or Shutdown.

Required?	false
Default Value	Nothing
Accept Pipeline Input?	true (ByPropertyName)

-OffPeakLogOffTimeout<Int32>

The number of minutes before the configured action should be performed after a user session ends outside peak hours.

Required?	false
Default Value	0
Accept Pipeline Input?	

Accept Pipeline Input?	true (ByPropertyName)

-PeakBufferSizePercent<Int32>

The percentage of machines in the desktop group that should be kept available in an idle state in peak hours.

Required?	false
Default Value	0
Accept Pipeline Input?	true (ByPropertyName)

-PeakDisconnectAction<SessionChangeHostingAction>

The action to be performed after a configurable period of a user session disconnecting in peak hours. Possible values are Nothing, Suspend, or Shutdown.

Required?	false
Default Value	Nothing
Accept Pipeline Input?	true (ByPropertyName)

-PeakDisconnectTimeout<Int32>

The number of minutes before the configured action should be performed after a user session disconnects in peak hours.

Required?	false
Default Value	0
Accept Pipeline Input?	true (ByPropertyName)

-PeakExtendedDisconnectAction<SessionChangeHostingAction>

The action to be performed after a second configurable period of a user session disconnecting in peak hours. Possible values are Nothing, Suspend, or Shutdown.

Required?	false
Default Value	Nothing

Accept Pipeline Input?	true (ByPropertyName)

-PeakExtendedDisconnectTimeout<Int32>

The number of minutes before the second configured action should be performed after a user session disconnects in peak hours.

Required?	false
Default Value	0
Accept Pipeline Input?	true (ByPropertyName)

-PeakLogOffAction<SessionChangeHostingAction>

The action to be performed after a configurable period of a user session ending in peak hours. Possible values are Nothing, Suspend, or Shutdown.

Required?	false
Default Value	Nothing
Accept Pipeline Input?	true (ByPropertyName)

-PeakLogOffTimeout<Int32>

The number of minutes before the configured action should be performed after a user session ends in peak hours.

Required?	false
Default Value	0
Accept Pipeline Input?	true (ByPropertyName)

-ProtocolPriority<String[]>

A list of protocol names in the order in which they should be attempted for use during connection.

Required?	false
Default Value	null

Accept Pipeline Input?	true (ByPropertyName)

-PublishedName<String>

The name that will be displayed to users for their desktop(s) in this desktop group.

Required?	false
Default Value	The same value as that supplied for the name of the desktop group.
Accept Pipeline Input?	true (ByPropertyName)

-Scope<String[]>

Specifies the name of the delegated administration scope to which the desktop group should belong.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-SecureIcaRequired<Boolean>

Whether HDX connections to desktops in the new desktop group require the use of a secure protocol.

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-SessionSupport<SessionSupport>

Specifies whether machines in the desktop group are single or multi-session capable. Values can be:

- o SingleSession - Single-session only machine.
- o MultiSession - Multi-session capable machine.

Required?	false
Default Value	

Accept Pipeline Input?	true (ByPropertyName)

-SettlementPeriodBeforeAutoShutdown<TimeSpan>

Time after a session ends during which automatic shutdown requests (for example, shutdown after use, idle pool management) are deferred. Any outstanding shutdown request takes effect after the settlement period expires. This is typically used to configure time to allow logoff scripts to complete.

Required?	false
Default Value	0
Accept Pipeline Input?	true (ByPropertyName)

-ShutdownDesktopsAfterUse<Boolean>

Whether desktops in this desktop group should be automatically shut down when each user session completes (only relevant to power-managed desktops).

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-TimeZone<String>

The time zone in which this desktop group's machines reside.

The time zone must be specified for any of the group's automatic power management settings to take effect. Automatic power management operations include pool management (power time schemes), reboot schedules, session disconnect and logoff actions, and powering on assigned machines etc.

Required?	false
Default Value	null
Accept Pipeline Input?	true (ByPropertyName)

-TurnOnAddedMachine<Boolean>

This flag specifies whether the Broker Service should attempt to power on machines when they are added to the desktop

group.

Required?	false
Default Value	\$false for single session machines and \$true for multi-session machines.
Accept Pipeline Input?	true (ByPropertyName)

-UUID<Guid>

An optional GUID for this desktop group.

Required?	false
Default Value	A new GUID is generated if none is supplied.
Accept Pipeline Input?	true (ByPropertyName)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.DesktopGroup

The newly created desktop group.

Notes

Once a new desktop group is created, you can create desktops in it by adding the appropriate broker machines to it using the Add-BrokerMachine or Add-BrokerMachinesToDesktopGroup cmdlets.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> New-BrokerDesktopGroup "Assigned Desktops" -PublishedName "MyDesktop" -DesktopKind Private
```

Create a desktop group to manage the brokering of private desktops, which will appear to users with the name "MyDesktop".

New-BrokerEntitlementPolicyRule

May 04, 2016
Creates a new desktop rule in the site's entitlement policy.

Syntax

New-BrokerEntitlementPolicyRule [-Name] <String> -DesktopGroupUid <Int32> [-ColorDepth <ColorDepth>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-IconUid <Int32>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-PublishedName <String>] [-SecureIcaRequired <Boolean>] [-SessionReconnection <SessionReconnection>] [-UUID <Guid>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Detailed Description

The New-BrokerEntitlementPolicyRule cmdlet adds a new desktop rule to the site's entitlement policy.

A desktop rule in the entitlement policy defines the users who are allowed per-session access to a machine from the rule's associated desktop group to run a full desktop session.

The following constraints apply when creating a desktop entitlement rule for a desktop group:

- o The group's desktop kind must be Shared
- o The group's delivery type must be DesktopsOnly or DesktopsAndApps

When a user selects a desktop entitlement published from a shared group, a machine is selected from the group on which to run the desktop session. No permanent association exists between the user and the selected machine; once the session ends the association also ends.

Multiple desktop rules in the entitlement policy can apply to the same desktop group. Where a user is granted an entitlement by more than one rule for the same group, they can use as many desktop sessions at the same time as they have entitlements.

Related topics

- [Get-BrokerEntitlementPolicyRule](#)
- [Set-BrokerEntitlementPolicyRule](#)
- [Rename-BrokerEntitlementPolicyRule](#)
- [Remove-BrokerEntitlementPolicyRule](#)

Parameters

-Name<String>

Specifies the administrative name of the new desktop rule. Each rule in the site's entitlement policy must have a unique name (irrespective of whether they are desktop or application rules).

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-DesktopGroupUid<Int32>

Specifies the unique ID of the desktop group to which the new desktop rule applies.

Required?	true

Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-ColorDepth<ColorDepth>

Specifies the color depth of any desktop sessions launched by a user from this entitlement.

Valid values are \$null, FourBit, EightBit, SixteenBit, and TwentyFourBit.

The default null value indicates that the equivalent setting from the rule's desktop group is used.

Required?	false
Default Value	null (dynamically inherited from the desktop group)
Accept Pipeline Input?	true (ByPropertyName)

-Description<String>

Specifies an optional description of the new desktop rule. The text may be visible to the end user, for example, as a tooltip associated with the desktop entitlement.

The default null value indicates that the equivalent setting from the rule's desktop group is used.

Required?	false
Default Value	null (dynamically inherited from the desktop group)
Accept Pipeline Input?	true (ByPropertyName)

-Enabled<Boolean>

Specifies whether the new desktop rule is initially enabled. A disabled rule is ignored when evaluating the site's entitlement policy.

Required?	false
Default Value	true
Accept Pipeline Input?	true (ByPropertyName)

-ExcludedUserFilterEnabled<Boolean>

Specifies whether the excluded users filter is initially enabled. If the filter is disabled then any user entries in the filter are ignored when entitlement policy rules are evaluated.

Required?	false
Default Value	false

Accept Pipeline Input?	true (ByPropertyName)

-ExcludedUsers<User[]>

Specifies the excluded users filter of the desktop rule, that is, the users and groups who are explicitly denied an entitlement to a desktop session from the new rule.

This can be used to exclude users or groups who would otherwise gain access by groups specified in the included users filter.

Required?	false
Default Value	(empty list)
Accept Pipeline Input?	true (ByPropertyName)

-IconUid<Int32>

Specifies the unique ID of the icon used to display the desktop session entitlement to the user.

The default null value indicates that the equivalent setting from the rule's desktop group is used.

Required?	false
Default Value	null (dynamically inherited from the desktop group)
Accept Pipeline Input?	true (ByPropertyName)

-IncludedUserFilterEnabled<Boolean>

Specifies whether the included users filter is initially enabled. If the filter is disabled then any user who satisfies the requirements of the access policy is implicitly granted an entitlement to a desktop session by the new rule.

Users who would be implicitly granted access when the filter is disabled can still be explicitly denied access using the excluded users filter.

Required?	false
Default Value	true
Accept Pipeline Input?	true (ByPropertyName)

-IncludedUsers<User[]>

Specifies the included users filter of the rule, that is, the users and groups who are granted an entitlement to a desktop session by the new rule.

If a user appears explicitly in the excluded users filter of the rule or is a member of a group that appears in the excluded users filter, no entitlement is granted whether or not the user appears in the included users filter.

Required?	false
Default Value	(empty list)
Accept Pipeline Input?	true (ByPropertyName)

-PublishedName<String>

The name of the new desktop session entitlement as seen by the user.

The default null value indicates that the equivalent setting from the rule's desktop group is used.

Required?	false
Default Value	null (dynamically inherited from the desktop group)
Accept Pipeline Input?	true (ByPropertyName)

-SecureIcaRequired<Boolean>

Specifies whether the new desktop rule requires the SecureICA protocol for desktop sessions launched using the entitlement.

The default null value indicates that the equivalent setting from the rule's desktop group is used.

Required?	false
Default Value	null (dynamically inherited from the desktop group)
Accept Pipeline Input?	true (ByPropertyName)

-SessionReconnection<SessionReconnection>

Defines reconnection (roaming) behavior for sessions launched using this rule. Session reconnection control is an experimental and unsupported feature.

Required?	false
Default Value	Always
Accept Pipeline Input?	true (ByPropertyName)

-UUID<Guid>

An optional GUID for this rule.

Required?	false
Default Value	A new GUID is generated if none is supplied.
Accept Pipeline Input?	true (ByPropertyName)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation

and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.EntitlementPolicyRule

New-BrokerEntitlementPolicyRule returns the newly created desktop rule in the entitlement policy.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> $dg = Get-BrokerDesktopGroup 'Customer Support'
```

```
C:\PS> New-BrokerEntitlementPolicyRule 'UK Office' -DesktopGroupUid $dg.Uid -IncludedUsers support\uk-staff -PublishedName 'Support Desktop'
```

Creates an desktop rule in the entitlement policy that entitles all members of the SUPPORT\uk-staff group to a desktop session from the Customer Support desktop group. The desktop entitlement name seen by users is Support Desktop.

New-BrokerHostingPowerAction

May 04, 2016

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

May 04, 2016
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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016
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

May 04, 2016
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

May 04, 2016
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

May 04, 2016

Create a new RemotePCAccount.

Syntax

```
New-BrokerRemotePCAccount -CatalogUid <Int32> -OU <String> [-AllowSubfolderMatches <Boolean>] [-MachinesExcluded <String[]>] [-MachinesIncluded <String[]>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Create a new RemotePCAccount. A RemotePCAccount defines machine filters to support Remote PC automation adding unconfigured machines to catalogs.

Related topics

[Get-BrokerRemotePCAccount](#)

[Set-BrokerRemotePCAccount](#)

[Remove-BrokerRemotePCAccount](#)

Parameters

-CatalogUid<Int32>

Specifies the catalog which Remote PC automation adds an unconfigured machine to if it matches this RemotePCAccount.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-OU<String>

Specifies the DN of an AD container, or has the special value 'any'.

When an AD container is specified a machine may only match with the RemotePCAccount when the AD computer object is located relative to the OU.

When 'any' is specified the location of the AD computer object is ignored for purposes of matching this RemotePCAccount. The machine must still meet the MachinesIncluded and MachinesExcluded filters for a match to occur.

In the event that a machine matches with multiple RemotePCAccounts then the RemotePCAccount OU with the longest canonical name takes precedence. The special 'any' OU is treated as lowest priority.

Note that the OU value of every RemotePCAccount must be unique, and this includes only one 'any' entry being permitted.

Required?	true
Default Value	null
Accept Pipeline Input?	true (ByPropertyName)

-AllowSubfolderMatches<Boolean>

When true a machine matches this RemotePCAccount if the AD computer object exists within the container specified by the OU property, or within a child container of the OU.

When false the AD computer object only matches if it exists directly in the AD container specified by the OU property.

This property is not meaningful when OU has the special value 'any'.

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-MachinesExcluded<String[]>

MachinesExcluded specifies a set of strings that can include asterisk wildcards. If a machine name matches any entries in MachinesExcluded then it cannot match with this RemotePCAccount regardless of whether there is a MachinesIncluded match.

Matches are performed against the domain name joined with the machine name by a backslash (DOMAIN\MACHINE), e.g.:

DOMAIN1\M*

DOMAIN*\M*

\M

Required?	false
Default Value	@()
Accept Pipeline Input?	true (ByPropertyName)

-MachinesIncluded<String[]>

MachinesIncluded specifies a set of strings that can include asterisk wildcards. A machine may only match with this RemotePCAccount if it matches a MachinesIncluded entry and does not match any MachinesExcluded entries.

Matches are performed against the domain name joined with the machine name by a backslash (DOMAIN\MACHINE), e.g.:

DOMAIN1\M*

DOMAIN*\M*

\M

Required?	false
Default Value	@('*')
Accept Pipeline Input?	true (ByPropertyName)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and

Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.RemotePCAccount

The newly created RemotePCAccount.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> New-BrokerRemotePCAccount -OU 'ou=MyOU,dc=MyDomain,dc=com' -CatalogUid 42
```

Create a RemotePCAccount that adds unconfigured machines with computer objects in MyOU, into catalog 42.

----- **EXAMPLE 2** -----

```
C:\PS> New-BrokerRemotePCAccount -OU 'any' -CatalogUid 42 -MachinesIncluded @('DOMAIN1\*') -MachinesExcluded @('DOMAIN1\*JOHNDOE*')
```

Create a RemotePCAccount matching unconfigured machines from DOMAIN1, except those with hostnames containing JOHNDOE, and add them to catalog 42.

----- **EXAMPLE 3** -----

```
C:\PS> New-BrokerRemotePCAccount -OU 'any' -CatalogUid 42
```

Create a RemotePCAccount that matches any unconfigured machine, causing automation to add matching machines to catalog 42.

New-BrokerSessionLinger

May 04, 2016

Creates a new session linger setting for a desktop group.

Syntax

New-BrokerSessionLinger [-DesktopGroupName] <String> [-Enabled <Boolean>] [-MaxAverageLoadThreshold <Int32>] [-MaxLoadPerMachineThreshold <Int32>] [-MaxTimeBeforeDisconnect <TimeSpan>] [-MaxTimeBeforeTerminate <TimeSpan>] [-UserFilterEnabled <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

New-BrokerSessionLinger -DesktopGroupUid <Int32> [-Enabled <Boolean>] [-MaxAverageLoadThreshold <Int32>] [-MaxLoadPerMachineThreshold <Int32>] [-MaxTimeBeforeDisconnect <TimeSpan>] [-MaxTimeBeforeTerminate <TimeSpan>] [-UserFilterEnabled <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Detailed Description

The New-BrokerSessionLinger cmdlet is used to define a session linger setting for a desktop group.

Note that each desktop group can only have a single session linger setting. Session lingering only applies to application sessions.

Related topics

[Get-BrokerSessionLinger](#)

[Set-BrokerSessionLinger](#)

[Remove-BrokerSessionLinger](#)

Parameters

-DesktopGroupName<String>

The name of the desktop group that this linger setting is applied to.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-DesktopGroupUid<Int32>

The Uid of the desktop group that this linger setting is applied to.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Enabled<Boolean>

Boolean that indicates if the new session linger is enabled.

Required?	false
Default Value	true
Accept Pipeline Input?	true (ByPropertyName)

-MaxAverageLoadThreshold<Int32>

Specifies the average load threshold across the desktop group. When the threshold hits, lingering sessions across the group be terminated to reduce load. Sessions that have been lingering the longest will be chosen first.

Required?	false
Default Value	0
Accept Pipeline Input?	true (ByPropertyName)

-MaxLoadPerMachineThreshold<Int32>

Specifies the maximum load threshold per machine in the desktop group. When the threshold hits, lingering sessions on each loaded machine will be terminated to reduce load. Sessions that have been lingering the longest will be chosen first.

Required?	false
Default Value	0
Accept Pipeline Input?	true (ByPropertyName)

-MaxTimeBeforeDisconnect<TimeSpan>

Specifies the time by which a lingering session will be disconnected. The disconnect timer is optional, but when specified the terminate timer needs to be also set. The disconnect time cannot be greater than the terminate time. When the disconnect and terminate times are the same, the terminate timer takes precedence. The disconnect timer needs to be paired with a session termination condition like the terminate timer or one of load threshold settings.

Required?	false
Default Value	15 minutes
Accept Pipeline Input?	true (ByPropertyName)

-MaxTimeBeforeTerminate<TimeSpan>

Specifies the time by which a lingering session will be terminated. The terminate timer is not optional when timers are configured. When the disconnect and terminate times are the same, the terminate timer takes precedence.

Required?	false

Default Value	8 hours
Accept Pipeline Input?	true (ByPropertyName)

-UserFilterEnabled<Boolean>

Specifies whether the session linger's user filter is enabled or disabled. Where the user filter is enabled, lingering is enabled only to users who appear in the filter (either explicitly or by virtue of group membership).

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Depends on parameter Parameters can be piped by property name.

Return Values

Citrix.Broker.Admin.SDK.SessionLinger

New-BrokerSessionLinger returns a session linger object.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> New-BrokerSessionLinger -DesktopGroupName test -Enabled $true -MaxTimeBeforeDisconnect 0:30 -MaxTimeBeforeTerminate 1:00
```

Creates a new session linger setting with a disconnect timer of 30 minutes and terminate timer of 1 hour.

New-BrokerSessionPreLaunch

May 04, 2016
Creates a new session pre-launch setting for a desktop group.

Syntax

New-BrokerSessionPreLaunch [-DesktopGroupName] <String> [-Enabled <Boolean>] [-MaxAverageLoadThreshold <Int32>] [-MaxLoadPerMachineThreshold <Int32>] [-MaxTimeBeforeDisconnect <TimeSpan>] [-MaxTimeBeforeTerminate <TimeSpan>] [-UserFilterEnabled <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

New-BrokerSessionPreLaunch -DesktopGroupUid <Int32> [-Enabled <Boolean>] [-MaxAverageLoadThreshold <Int32>] [-MaxLoadPerMachineThreshold <Int32>] [-MaxTimeBeforeDisconnect <TimeSpan>] [-MaxTimeBeforeTerminate <TimeSpan>] [-UserFilterEnabled <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Detailed Description

The New-BrokerSessionPreLaunch cmdlet is used to define a session pre-launch setting for a desktop group.

Note that each desktop group can only have a single session pre-launch setting. Session pre-launch only applies to application sessions.

Related topics

- [Get-BrokerSessionPreLaunch](#)
- [Set-BrokerSessionPreLaunch](#)
- [Remove-BrokerSessionPreLaunch](#)

Parameters

-DesktopGroupName<String>

The name of the desktop group that this pre-launch setting is applied to.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-DesktopGroupUid<Int32>

The Uid of the desktop group that this pre-launch setting is applied to.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Enabled<Boolean>

Boolean that indicates if the new session pre-launch is enabled.

Required?	false
Default Value	true
Accept Pipeline Input?	true (ByPropertyName)

-MaxAverageLoadThreshold<Int32>

Specifies the average load threshold across the desktop group. When the threshold hits, pre-launched sessions across the group be terminated to reduce load. Sessions that have been pre-launched the longest will be chosen first.

Required?	false
Default Value	0
Accept Pipeline Input?	true (ByPropertyName)

-MaxLoadPerMachineThreshold<Int32>

Specifies the maximum load threshold per machine in the desktop group. When the threshold hits, pre-launched sessions on each loaded machine will be terminated to reduce load. Sessions that have been pre-launched the longest will be chosen first.

Required?	false
Default Value	0
Accept Pipeline Input?	true (ByPropertyName)

-MaxTimeBeforeDisconnect<TimeSpan>

Specifies the time by which a pre-launched session will be disconnected. The disconnect timer is optional, but when specified the terminate timer needs to be also set. The disconnect time cannot be greater than the terminate time. When the disconnect and terminate times are the same, the terminate timer takes precedence. The disconnect timer needs to be paired with a session termination condition like the terminate timer or one of load threshold settings.

Required?	false
Default Value	15 minutes
Accept Pipeline Input?	true (ByPropertyName)

-MaxTimeBeforeTerminate<TimeSpan>

Specifies the time by which a pre-launched session will be terminated. The terminate timer is not optional when timers are configured. When the disconnect and terminate times are the same, the terminate timer takes precedence.

Required?	false
Default Value	8 hours

Accept Pipeline Input?	true (ByPropertyName)

-UserFilterEnabled<Boolean>

Specifies whether the session pre-launch's user filter is enabled or disabled. Where the user filter is enabled, pre-launch is enabled only to users who appear in the filter (either explicitly or by virtue of group membership).

Required?	false
Default Value	false
Accept Pipeline Input?	true (ByPropertyName)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Depends on parameter Parameters can be piped by property name.

Return Values

Citrix.Broker.Admin.SDK.SessionPreLaunch

New-BrokerSessionPreLaunch returns a session pre-launch object.

Examples

----- **EXAMPLE 1** -----

C:\PS> New-BrokerSessionPreLaunch -DesktopGroupName test -Enabled \$true -MaxTimeBeforeDisconnect 0:30 -MaxTimeBeforeTerminate 1:00
Creates a new session pre-launch setting with a disconnect timer of 30 minutes and terminate timer of 1 hour.

New-BrokerTag

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Deletes AccessPolicyRule Metadata from the AccessPolicyRule objects

Syntax

```
Remove-BrokerAccessPolicyRuleMetadata [-InputObject] <AccessPolicyRule[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-BrokerAccessPolicyRuleMetadata cmdlet deletes Metadata from the AccessPolicyRule objects.

Related topics

Parameters

-InputObject<AccessPolicyRule[]>

Specifies the AccessPolicyRule object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerAccessPolicyRule You can pipe the AccessPolicyRule to delete the metadata.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerAccessPolicyRuleMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the AccessPolicyRule whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerAccessPolicyRule | Remove-BrokerAccessPolicyRuleMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the AccessPolicyRule in the site

Remove-BrokerAdminFolder

May 04, 2016

Removes an admin folder.

Syntax

```
Remove-BrokerAdminFolder [-InputObject] <AdminFolder[]> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-BrokerAdminFolder [-Name] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-BrokerAdminFolder cmdlet removes an existing admin folder.

Remove-BrokerAdminFolder will not remove a folder if it contains any other objects (e.g. sub-folders or applications).

Related topics

[Get-BrokerAdminFolder](#)

[New-BrokerAdminFolder](#)

Parameters

-InputObject<AdminFolder[]>

Identifies the folder to remove

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

The name pattern of folder(s) to remove

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.AdminFolder The admin folder objects can be specified as input.

Return Values

None

This cmdlet does not return any output.

Examples

----- **EXAMPLE 1** -----

```
Remove-BrokerAdminFolder F1\F2\
```

Removes the folder called F2 within the folder F1\

Remove-BrokerAdminFolderMetadata

May 04, 2016

Deletes AdminFolder Metadata from the AdminFolder objects

Syntax

```
Remove-BrokerAdminFolderMetadata [-InputObject] <AdminFolder[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-BrokerAdminFolderMetadata cmdlet deletes Metadata from the AdminFolder objects.

Related topics

Parameters

-InputObject<AdminFolder[]>

Specifies the AdminFolder object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerAdminFolder You can pipe the AdminFolder to delete the metadata.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerAdminFolderMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the AdminFolder whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerAdminFolder | Remove-BrokerAdminFolderMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the AdminFolder in the site

Remove-BrokerAppAssignmentPolicyRule

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Deletes one or more applications, or an association of an application.

Syntax

```
Remove-BrokerApplication [-InputObject] <Application[]> [-Force] [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerApplication [-Name] <String> [-Force] [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-BrokerApplication cmdlet deletes one or more applications, or you can use it to delete just the association of an application to a desktop group.

Its usage dictates the behavior of the cmdlet. If only the application is specified as a parameter, then the cmdlet deletes the application. It also deletes any associations this application has with other objects, such as with access policy rules or desktop groups. More specifically, when an application is deleted the following happens:

- o The association to any desktop groups is removed.
- o The association to any tags is removed.
- o Any configured file-type association objects for this application are deleted.
- o The association to any user accounts is removed.
- o The association to any access session conditions is removed.
- o The access policy rule object for this application, if one existed, is deleted.
- o Finally, the application object itself is deleted.

Note that if the application is in use by a user then the application cannot be deleted.

If more than just the application is supplied as a parameter to the cmdlet (for instance, if a DesktopGroup object is also specified) then the application is not deleted. Instead, only the association from the application to that desktop group is removed.

Related topics

[New-BrokerApplication](#)

[Add-BrokerApplication](#)

[Get-BrokerApplication](#)

[Rename-BrokerApplication](#)

[Move-BrokerApplication](#)

Set-BrokerApplication

Parameters

-InputObject<Application[]>

Specifies the applications to delete. The Uid can also be substituted for the application objects.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the application to remove.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Force<SwitchParameter>

Remove application even if it's in use. Removing an application that is currently in use, can potentially leave an application session containing no applications. If all the applications that are currently active in a disconnected application session are removed, the user will be unable to reconnect to the session. Forcing removal of an in-use application does not impact the actual session itself.

Required?	false
Default Value	false
Accept Pipeline Input?	false

-DesktopGroup<DesktopGroup>

Specifies the desktop group that this application should no longer be associated with. The Uid or Name can also be substituted for the desktop group object.

Required?	false

Default Value	
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.Application The application objects can be specified as input.

Return Values

None

This cmdlet does not return any output.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerApplication "Notepad"
```

This command deletes the application that has a BrowserName of "Notepad".

----- EXAMPLE 2 -----

```
C:\PS> $app = Get-BrokerApplication -BrowserName "Notepad"
```

```
C:\PS> $group = Get-BrokerDesktopGroup -Name "Private DesktopGroup"
```

```
C:\PS> Remove-BrokerApplication -InputObject $app -DesktopGroup $group
```

This command removes the association of the desktop group that has a name of "Private DesktopGroup" from the application that has a BrowserName of "Notepad". It does not otherwise modify the application.

Remove-BrokerApplicationInstanceMetadata

May 04, 2016

Deletes ApplicationInstance Metadata from the ApplicationInstance objects

Syntax

```
Remove-BrokerApplicationInstanceMetadata [-InputObject] <ApplicationInstance[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-BrokerApplicationInstanceMetadata cmdlet deletes Metadata from the ApplicationInstance objects.

Related topics

Parameters

-InputObject<ApplicationInstance[]>

Specifies the ApplicationInstance object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerApplicationInstance You can pipe the ApplicationInstance to delete the metadata.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerApplicationInstanceMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the ApplicationInstance whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerApplicationInstance | Remove-BrokerApplicationInstanceMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the ApplicationInstance in the site

Remove-BrokerApplicationMetadata

May 04, 2016

Deletes Application Metadata from the Application objects

Syntax

```
Remove-BrokerApplicationMetadata [-InputObject] <Application[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-BrokerApplicationMetadata cmdlet deletes Metadata from the Application objects.

Related topics

Parameters

-InputObject<Application[]>

Specifies the Application object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerApplication You can pipe the Application to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerApplicationMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the Application whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerApplication | Remove-BrokerApplicationMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the Application in the site

Remove-BrokerAssignmentPolicyRule

May 04, 2016

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

May 04, 2016

Deletes AssignmentPolicyRule Metadata from the AssignmentPolicyRule objects

Syntax

```
Remove-BrokerAssignmentPolicyRuleMetadata [-InputObject] <AssignmentPolicyRule[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-BrokerAssignmentPolicyRuleMetadata cmdlet deletes Metadata from the AssignmentPolicyRule objects.

Related topics

Parameters

-InputObject<AssignmentPolicyRule[]>

Specifies the AssignmentPolicyRule object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerAssignmentPolicyRule You can pipe the AssignmentPolicyRule to delete the metadata.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerAssignmentPolicyRuleMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the AssignmentPolicyRule whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerAssignmentPolicyRule | Remove-BrokerAssignmentPolicyRuleMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the AssignmentPolicyRule in the site

Remove-BrokerCatalog

May 04, 2016

Removes catalogs from the site.

Syntax

```
Remove-BrokerCatalog [-InputObject] <Catalog[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerCatalog [-Name] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Remove catalogs from the site.

In order to remove a catalog from a site, the catalog must not contain machines. To remove a machine from a catalog use the Remove-BrokerMachine cmdlet. Note: in order to remove a machine from a catalog, it must not belong to a desktop group.

Related topics

[New-BrokerCatalog](#)

[Get-BrokerCatalog](#)

[Rename-BrokerCatalog](#)

[Set-BrokerCatalog](#)

[New-BrokerDesktopGroup](#)

[Remove-BrokerDesktopGroup](#)

Parameters

-InputObject<Catalog[]>

Specifies the catalog objects to delete.

Required?	true
Default Value	null
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the catalog to delete.

Required?	true

Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.Catalog You can pipe the catalogs to be deleted to Remove-BrokerCatalog.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerCatalog -Name "MyCatalog"
C:\PS> Remove-BrokerCatalog -InputObject (Get-BrokerCatalog -Name "MyCatalog")
```

These commands delete the catalog with the name "MyCatalog".

----- EXAMPLE 2 -----


```
C:\PS> Remove-BrokerCatalog -Name 'test'
```

This command deletes all catalogs with names beginning with "test".

----- **EXAMPLE 3** -----

```
C:\PS> Get-BrokerCatalog -RemotePCDesktopGroupUid 42 | Remove-BrokerCatalog -RemotePCDesktopGroup 42
```

Remove all the Remote PC catalogs that are associated with desktop group 42. Note that this only breaks the Remote PC relationships and does not delete the desktop groups.

Remove-BrokerCatalogMetadata

May 04, 2016

Deletes Catalog Metadata from the Catalog objects

Syntax

```
Remove-BrokerCatalogMetadata [-InputObject] <Catalog[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

The Remove-BrokerCatalogMetadata cmdlet deletes Metadata from the Catalog objects.

Related topics

Parameters

-InputObject<Catalog[]>

Specifies the Catalog object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerCatalog You can pipe the Catalog to delete the metadata.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerCatalogMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the Catalog whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerCatalog | Remove-BrokerCatalogMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the Catalog in the site

Remove-BrokerConfigurationSlot

May 04, 2016

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

May 04, 2016

Deletes ConfigurationSlot Metadata from the ConfigurationSlot objects

Syntax

```
Remove-BrokerConfigurationSlotMetadata [-InputObject] <ConfigurationSlot[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-BrokerConfigurationSlotMetadata cmdlet deletes Metadata from the ConfigurationSlot objects.

Related topics

Parameters

-InputObject<ConfigurationSlot[]>

Specifies the ConfigurationSlot object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerConfigurationSlot You can pipe the ConfigurationSlot to delete the metadata.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerConfigurationSlotMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the ConfigurationSlot whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerConfigurationSlot | Remove-BrokerConfigurationSlotMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the ConfigurationSlot in the site

Remove-BrokerConfiguredFTA

May 04, 2016

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

May 04, 2016

Deletes Controller Metadata from the Controller objects

Syntax

```
Remove-BrokerControllerMetadata [-InputObject] <Controller[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-BrokerControllerMetadata cmdlet deletes Metadata from the Controller objects.

Related topics

Parameters

-InputObject<Controller[]>

Specifies the Controller object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerController You can pipe the Controller to delete the metadata.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerControllerMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the Controller whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerController | Remove-BrokerControllerMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the Controller in the site

Remove-BrokerDelayedHostingPowerAction

May 04, 2016

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

May 04, 2016

Remove desktop groups from the system or remove them from a Remote PC catalog.

Syntax

```
Remove-BrokerDesktopGroup [-InputObject] <DesktopGroup[]> [-Force] [-RemotePCCatalog <Catalog>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerDesktopGroup [-Name] <String> [-Force] [-RemotePCCatalog <Catalog>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

This cmdlet has 2 functions:

- o Remove desktop groups from the system.
- o Break Remote PC associations between desktop groups and a catalog.

The Remote PC relationships are used by Remote PC automation to determine which desktop groups a machine in a particular Remote PC catalog can be published to. The assignment policy rules belonging to those desktop groups also determines the set of users that are allowed to be assigned to machines from the catalog.

Related topics

[Get-BrokerDesktopGroup](#)

[New-BrokerDesktopGroup](#)

[Set-BrokerDesktopGroup](#)

[Add-BrokerDesktopGroup](#)

[Rename-BrokerDesktopGroup](#)

[New-BrokerCatalog](#)

[Remove-BrokerCatalog](#)

Parameters

-InputObject <DesktopGroup[]>

Specifies the desktop groups to remove.

Required?	true
Default Value	null
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the desktop group to remove.

Required?	true
Default Value	null
Accept Pipeline Input?	true (ByPropertyName)

-Force<SwitchParameter>

Remove desktop groups even if there are active sessions.

Required?	false
Default Value	false
Accept Pipeline Input?	false

-RemotePCCatalog<Catalog>

When this parameter is specified, Remote PC desktop groups are removed from the specified Remote PC catalog.

Required?	false
Default Value	null
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	

Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.DesktopGroup You can pipe desktop groups to Remove-BrokerDesktopGroup.

Return Values

None

Notes

If a desktop group contains desktops when it is removed, these desktops are also removed (but the underlying broker machine remains).

A desktop group that still has active sessions cannot be removed unless the -Force switch is used.

Examples

----- **EXAMPLE 1** -----

C:\PS> Remove-BrokerDesktopGroup EMEA*
Remove all desktop groups with names starting with "EMEA".

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerDesktopGroup -Enabled \$false | Remove-BrokerDesktopGroup -Force
Remove all desktops that are currently disabled even if there are active sessions.

----- **EXAMPLE 3** -----

C:\PS> Get-BrokerDesktopGroup -RemotePCCatalogUid 42 | Remove-BrokerDesktopGroup -RemotePCCatalog 42
Remove all the Remote PC desktop groups that are associated with catalog 42. Note that this only breaks the Remote PC relationships and does not delete the desktop groups.

Remove-BrokerDesktopGroupMetadata

May 04, 2016

Deletes DesktopGroup Metadata from the DesktopGroup objects

Syntax

```
Remove-BrokerDesktopGroupMetadata [-InputObject] <DesktopGroup[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-BrokerDesktopGroupMetadata cmdlet deletes Metadata from the DesktopGroup objects.

Related topics

Parameters

-InputObject<DesktopGroup[]>

Specifies the DesktopGroup object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerDesktopGroup You can pipe the DesktopGroup to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerDesktopGroupMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the DesktopGroup whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerDesktopGroup | Remove-BrokerDesktopGroupMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the DesktopGroup in the site

Remove-BrokerEntitlementPolicyRule

May 04, 2016

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

May 04, 2016

Deletes EntitlementPolicyRule Metadata from the EntitlementPolicyRule objects

Syntax

```
Remove-BrokerEntitlementPolicyRuleMetadata [-InputObject] <EntitlementPolicyRule[]> -Name <String> [-LoggingId <Guid>]  
[-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-BrokerEntitlementPolicyRuleMetadata cmdlet deletes Metadata from the EntitlementPolicyRule objects.

Related topics

Parameters

-InputObject<EntitlementPolicyRule[]>

Specifies the EntitlementPolicyRule object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerEntitlementPolicyRule You can pipe the EntitlementPolicyRule to delete the metadata.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerEntitlementPolicyRuleMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the EntitlementPolicyRule whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerEntitlementPolicyRule | Remove-BrokerEntitlementPolicyRuleMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the EntitlementPolicyRule in the site

Remove-BrokerHostingPowerAction

May 04, 2016

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

May 04, 2016

Deletes HostingPowerAction Metadata from the HostingPowerAction objects

Syntax

```
Remove-BrokerHostingPowerActionMetadata [-InputObject] <HostingPowerAction[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-BrokerHostingPowerActionMetadata cmdlet deletes Metadata from the HostingPowerAction objects.

Related topics

Parameters

-InputObject<HostingPowerAction[]>

Specifies the HostingPowerAction object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerHostingPowerAction You can pipe the HostingPowerAction to delete the metadata.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerHostingPowerActionMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the HostingPowerAction whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerHostingPowerAction | Remove-BrokerHostingPowerActionMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the HostingPowerAction in the site

Remove-BrokerHypervisorAlertMetadata

May 04, 2016

Deletes HypervisorAlert Metadata from the HypervisorAlert objects

Syntax

```
Remove-BrokerHypervisorAlertMetadata [-InputObject] <HypervisorAlert[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-BrokerHypervisorAlertMetadata cmdlet deletes Metadata from the HypervisorAlert objects.

Related topics

Parameters

-InputObject<HypervisorAlert[]>

Specifies the HypervisorAlert object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerHypervisorAlert You can pipe the HypervisorAlert to delete the metadata.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerHypervisorAlertMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the HypervisorAlert whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerHypervisorAlert | Remove-BrokerHypervisorAlertMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the HypervisorAlert in the site

Remove-BrokerHypervisorConnection

May 04, 2016

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

May 04, 2016

Deletes HypervisorConnection Metadata from the HypervisorConnection objects

Syntax

```
Remove-BrokerHypervisorConnectionMetadata [-InputObject] <HypervisorConnection[]> -Name <String> [-LoggingId <Guid>]  
[-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-BrokerHypervisorConnectionMetadata cmdlet deletes Metadata from the HypervisorConnection objects.

Related topics

Parameters

-InputObject<HypervisorConnection[]>

Specifies the HypervisorConnection object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerHypervisorConnection You can pipe the HypervisorConnection to delete the metadata.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerHypervisorConnectionMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the HypervisorConnection whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerHypervisorConnection | Remove-BrokerHypervisorConnectionMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the HypervisorConnection in the site

Remove-BrokerIcon

May 04, 2016

Remove an icon.

Syntax

```
Remove-BrokerIcon [-InputObject] <Icon[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Removes an icon from the database.

Related topics

[Get-BrokerIcon](#)

[New-BrokerIcon](#)

[Set-BrokerIconMetadata](#)

Parameters

-InputObject<Icon[]>

Specifies the icon to remove.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.Icon The icon to be removed can be piped into the cmdlet.

Return Values

None

Notes

Note that if the icon is currently in use, for example, as a desktop icon, it cannot be removed until the association is cleared.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerIcon 3  
Removes the icon with Uid 3.
```

Remove-BrokerIconMetadata

May 04, 2016

Deletes Icon Metadata from the Icon objects

Syntax

```
Remove-BrokerIconMetadata [-InputObject] <Icon[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

The Remove-BrokerIconMetadata cmdlet deletes Metadata from the Icon objects.

Related topics

Parameters

-InputObject<Icon[]>

Specifies the Icon object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerIcon You can pipe the Icon to delete the metadata.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerIconMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the Icon whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerIcon | Remove-BrokerIconMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the Icon in the site

Remove-BrokerImportedFTA

May 04, 2016

Deletes one or more imported file type associations.

Syntax

```
Remove-BrokerImportedFTA -DesktopGroupUids <Int32[]> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

Deletes all of the imported file type associations belonging to one or more desktop groups. At least one desktop group must be specified.

Imported file type associations are grouped together based on the desktop group of the machine from which they were imported. All file types for a desktop group are deleted. There is no mechanism for deleting a subset imported file type associations for a specific desktop group.

Imported file type associations are different from configured file type associations. Imported file type associations are lists of known file type associations for a given desktop group. Configured file type associations are those that are actually associated with published applications for the purposes of content redirection.

Related topics

[Get-BrokerImportedFTA](#)

[Update-BrokerImportedFTA](#)

Parameters

-DesktopGroupUids<Int32[]>

Deletes the imported file type associations belonging to specified desktop groups.

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Int32[] An array of Uids for the desktop groups can be supplied as input. The desktop groups must be of the Private or Shared desktop kind.

Return Values

None

Notes

If an imported file type association is used to create a new configured file type association and the imported file type association is subsequently deleted, the configured file type association is not affected.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> $dg = Get-BrokerDesktopGroup -Name "Sales VMs"
C:\PS> Remove-BrokerImportedFTA -DesktopGroupUids $dg.Uid
```

Deletes all imported file type associations belonging to the "Sales VMs" desktop group.

Remove-BrokerLease

May 04, 2016

Remove the specified lease in the Database.

Syntax

Remove-BrokerLease [-InputObject] <Lease[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Remove-BrokerLease [-Key] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Detailed Description

Marks the specified lease for deletion. Note that the lease is eventually fully deleted when enough time has been allowed for the deletion to propagate to all controller machines in the site, but is immediately removed from lease search results.

Related topics

[Update-BrokerLocalLeaseCache](#)

Remove-BrokerCache

Parameters

-InputObject<Lease[]>

Specifies the lease to remove.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Key<String>

Specifies the lease key of the lease to remove. A pattern can be specified.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop

Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.Lease The lease to be removed can be piped into the cmdlet.

Return Values

None

This cmdlet does not return any output.

Notes

The lease is marked for deletion after this cmdlet is run. Note that the lease is eventually fully deleted when enough time has been allowed for the deletion to propagate to all controller machines in the site, but is immediately removed from lease search results.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> $lease = Get-BrokerLease -Uid 1
```

```
C:\PS> Remove-BrokerLease $lease
```

Marks the specified lease for deletion.

Remove-BrokerLeaseMetadata

May 04, 2016

Deletes Lease Metadata from the Lease objects

Syntax

```
Remove-BrokerLeaseMetadata [-InputObject] <Lease[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

The Remove-BrokerLeaseMetadata cmdlet deletes Metadata from the Lease objects.

Related topics

Parameters

-InputObject<Lease[]>

Specifies the Lease object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerLease You can pipe the Lease to delete the metadata.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerLeaseMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the Lease whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerLease | Remove-BrokerLeaseMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the Lease in the site

Remove-BrokerMachine

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Deletes MachineCommand Metadata from the MachineCommand objects

Syntax

```
Remove-BrokerMachineCommandMetadata [-InputObject] <MachineCommand[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-BrokerMachineCommandMetadata cmdlet deletes Metadata from the MachineCommand objects.

Related topics

Parameters

-InputObject<MachineCommand[]>

Specifies the MachineCommand object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerMachineCommand You can pipe the MachineCommand to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerMachineCommandMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the MachineCommand whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerMachineCommand | Remove-BrokerMachineCommandMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the MachineCommand in the site

Remove-BrokerMachineConfiguration

May 04, 2016

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

May 04, 2016

Deletes MachineConfiguration Metadata from the MachineConfiguration objects

Syntax

```
Remove-BrokerMachineConfigurationMetadata [-InputObject] <MachineConfiguration[]> -Name <String> [-LoggingId <Guid>]  
[-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-BrokerMachineConfigurationMetadata cmdlet deletes Metadata from the MachineConfiguration objects.

Related topics

Parameters

-InputObject<MachineConfiguration[]>

Specifies the MachineConfiguration object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerMachineConfiguration You can pipe the MachineConfiguration to delete the metadata.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerMachineConfigurationMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the MachineConfiguration whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerMachineConfiguration | Remove-BrokerMachineConfigurationMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the MachineConfiguration in the site

Remove-BrokerMachineMetadata

May 04, 2016

Deletes Machine Metadata from the Machine objects

Syntax

```
Remove-BrokerMachineMetadata [-InputObject] <Machine[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

The Remove-BrokerMachineMetadata cmdlet deletes Metadata from the Machine objects.

Related topics

Parameters

-InputObject<Machine[]>

Specifies the Machine object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerMachine You can pipe the Machine to delete the metadata.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerMachineMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the Machine whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerMachine | Remove-BrokerMachineMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the Machine in the site

Remove-BrokerPowerTimeScheme

May 04, 2016
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

May 04, 2016

Deletes PowerTimeScheme Metadata from the PowerTimeScheme objects

Syntax

```
Remove-BrokerPowerTimeSchemeMetadata [-InputObject] <PowerTimeScheme[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-BrokerPowerTimeSchemeMetadata cmdlet deletes Metadata from the PowerTimeScheme objects.

Related topics

Parameters

-InputObject<PowerTimeScheme[]>

Specifies the PowerTimeScheme object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerPowerTimeScheme You can pipe the PowerTimeScheme to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerPowerTimeSchemeMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the PowerTimeScheme whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerPowerTimeScheme | Remove-BrokerPowerTimeSchemeMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the PowerTimeScheme in the site

Remove-BrokerRebootCycleMetadata

May 04, 2016

Deletes RebootCycle Metadata from the RebootCycle objects

Syntax

```
Remove-BrokerRebootCycleMetadata [-InputObject] <RebootCycle[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-BrokerRebootCycleMetadata cmdlet deletes Metadata from the RebootCycle objects.

Related topics

Parameters

-InputObject<RebootCycle[]>

Specifies the RebootCycle object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerRebootCycle You can pipe the RebootCycle to delete the metadata.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerRebootCycleMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the RebootCycle whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerRebootCycle | Remove-BrokerRebootCycleMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the RebootCycle in the site

Remove-BrokerRebootSchedule

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Remove the specified catalog/desktop group from the given scope(s).

Syntax

```
Remove-BrokerScope [-InputObject] <Scope[]> [-Catalog <Catalog>] [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-BrokerScope cmdlet is used to remove a catalog/desktop group object from the given scope(s).

To remove a catalog/desktop group from a scope you need permission to change the scopes of the catalog/desktop group.

If the catalog/desktop group is not in a specified scope, that scope will be silently ignored.

Related topics

Parameters

-InputObject<Scope[]>

Specifies the scopes to remove the object from.

Required?	true
Default Value	null
Accept Pipeline Input?	true (ByValue)

-Catalog<Catalog>

Specifies the catalog object to be removed.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByValue)

-DesktopGroup<DesktopGroup>

Specifies the desktop group object to be removed.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.Scope You can pipe scopes to Remove-BrokerScope.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerScope -Scope Chalfont,Cambourne -DesktopGroup "Win7 Desktops"
```

Removes the "Win7 Desktops" desktop group from both the Chalfont and Cambourne scopes.

----- EXAMPLE 2 -----

```
C:\PS> 'Chalfont','Cambourne' | Remove-BrokerScope -DesktopGroup 'Win7 Desktops'
```

Removes the "Win7 Desktops" desktop group from both the Chalfont and Cambourne scopes.

Remove-BrokerSessionLinger

May 04, 2016

Removes a session linger setting.

Syntax

```
Remove-BrokerSessionLinger [-InputObject] <SessionLinger[]> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-BrokerSessionLinger [-DesktopGroupName] <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

The Remove-BrokerSessionLinger cmdlet is used to delete an existing session linger setting.

Related topics

[New-BrokerSessionLinger](#)

[Get-BrokerSessionLinger](#)

[Set-BrokerSessionLinger](#)

Parameters

-InputObject<SessionLinger[]>

The session linger setting to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-DesktopGroupName<String>

The name of the desktop group whose session linger setting is to be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.SessionLinger Session linger settings may be specified through pipeline input.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Get-BrokerSessionLinger | Remove-BrokerSessionLinger
```

Deletes every session linger setting for all desktop groups.

Remove-BrokerSessionMetadata

May 04, 2016

Deletes Session Metadata from the Session objects

Syntax

```
Remove-BrokerSessionMetadata [-InputObject] <Session[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

The Remove-BrokerSessionMetadata cmdlet deletes Metadata from the Session objects.

Related topics

Parameters

-InputObject<Session[]>

Specifies the Session object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerSession You can pipe the Session to delete the metadata.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerSessionMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the Session whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerSession | Remove-BrokerSessionMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the Session in the site

Remove-BrokerSessionPreLaunch

May 04, 2016

Removes a session pre-launch setting.

Syntax

```
Remove-BrokerSessionPreLaunch [-InputObject] <SessionPreLaunch[]> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-BrokerSessionPreLaunch [-DesktopGroupName] <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

The Remove-BrokerSessionPreLaunch cmdlet is used to delete an existing session pre-launch setting.

Related topics

[New-BrokerSessionPreLaunch](#)

[Get-BrokerSessionPreLaunch](#)

[Set-BrokerSessionPreLaunch](#)

Parameters

-InputObject<SessionPreLaunch[]>

The session pre-launch setting to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-DesktopGroupName<String>

The name of the desktop group whose session pre-launch setting is to be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.SessionPreLaunch Session pre-launch settings may be specified through pipeline input.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Get-BrokerSessionPreLaunch | Remove-BrokerSessionPreLaunch
```

Deletes every session pre-launch setting for all desktop groups.

Remove-BrokerSiteMetadata

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Deletes Tag Metadata from the Tag objects

Syntax

```
Remove-BrokerTagMetadata [-InputObject] <Tag[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

The Remove-BrokerTagMetadata cmdlet deletes Metadata from the Tag objects.

Related topics

Parameters

-InputObject<Tag[]>

Specifies the Tag object's instance whose Metadata is to be deleted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata to be deleted

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerTag You can pipe the Tag to delete the metadata.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerTagMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the Tag whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerTag | Remove-BrokerTagMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the Tag in the site

Remove-BrokerUser

May 04, 2016

Remove broker user objects from another broker object

Syntax

```
Remove-BrokerUser [-InputObject] <User[]> [-Application <Application>] [-SessionLinger <SessionLinger>] [-SessionPreLaunch <SessionPreLaunch>] [-Machine <Machine>] [-PrivateDesktop <PrivateDesktop>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerUser [-Name] <String> [-Application <Application>] [-SessionLinger <SessionLinger>] [-SessionPreLaunch <SessionPreLaunch>] [-Machine <Machine>] [-PrivateDesktop <PrivateDesktop>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-BrokerUser cmdlet removes broker user objects from another specified object, such as a broker private desktop, to which the user had previously been added.

Related topics

[Add-BrokerUser](#)

[Get-BrokerUser](#)

Parameters

-InputObject<User[]>

Specifies the user objects to remove.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the user objects to remove, based on their Name property.

Required?	true
Default Value	null
Accept Pipeline Input?	true (ByPropertyName)

-Application<Application>

The application from which to remove the user

Required?	false
Default Value	null
Accept Pipeline Input?	true (ByValue)

-SessionLinger<SessionLinger>

The desktop group session linger setting from which to remove the user.

Required?	false
Default Value	null
Accept Pipeline Input?	true (ByValue)

-SessionPreLaunch<SessionPreLaunch>

The desktop group session pre-launch setting from which to remove the user.

Required?	false
Default Value	null
Accept Pipeline Input?	true (ByValue)

-Machine<Machine>

The machine from which to remove the user

Required?	false
Default Value	null
Accept Pipeline Input?	true (ByValue)

-PrivateDesktop<PrivateDesktop>

The desktop from which to remove the user

Required?	false
Default Value	null
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.User You can pipe the users to be removed to Remove-BrokerUser.

Return Values

None

Notes

Specify one of the -Machine or -PrivateDesktop parameters only.

Examples

----- **EXAMPLE 1** -----

```
Remove-BrokerUser "DOMAIN\UserName" -PrivateDesktop "DOMAIN\MachineName"
```

Remove the assignment of the specified private desktop to the specified user.

Rename-BrokerAccessPolicyRule

May 04, 2016
Renames a rule in the site's access policy.

Syntax

Rename-BrokerAccessPolicyRule [-InputObject] <AccessPolicyRule[]> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Rename-BrokerAccessPolicyRule [-Name] <String> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Detailed Description

The Rename-BrokerAccessPolicyRule cmdlet renames a rule in the site's access policy. The Name property of the rule is changed.

An access policy rule defines a set of connection filters and access control rights relating to a desktop group. These allow fine-grained control of what access is granted to a desktop group based on details of, for example, a user's endpoint device, its address, and the user's identity.

Related topics

- [New-BrokerAccessPolicyRule](#)
- [Get-BrokerAccessPolicyRule](#)
- [Set-BrokerAccessPolicyRule](#)
- [Remove-BrokerAccessPolicyRule](#)

Parameters

-InputObject<AccessPolicyRule[]>

The access policy rule to be renamed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

The existing name of the access policy rule to be renamed.

Required?	true

Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-NewName<String>

The new name for the access policy rule being renamed. The new name must not match that of any other existing rules in the policy.

Required?	true
Default Value	
Accept Pipeline Input?	false

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host

name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.AccessPolicyRule The access policy rule to be renamed.

Return Values

None or Citrix.Broker.Admin.SDK.AccessPolicyRule

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.AccessPolicyRule object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Rename-BrokerAccessPolicyRule 'Sales' -NewName 'TeleSales'
```

Renames the access policy rule called Sales to TeleSales. The new name of the rule must be unique in the access policy.

Rename-BrokerAdminFolder

May 04, 2016

Renames a folder

Syntax

```
Rename-BrokerAdminFolder [-InputObject] <AdminFolder[]> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Rename-BrokerAdminFolder [-Name] <String> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Rename-BrokerAdminFolder cmdlet renames a folder for organising objects for administration purposes (for example, Applications).

The following special characters are not allowed in the new FolderName: \ / ; : # . * ? = < > | [] () " ' `

Related topics

[Get-BrokerAdminFolder](#)

[New-BrokerAdminFolder](#)

Parameters

-InputObject<AdminFolder[]>

The folder(s) to be renamed

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

A pattern matching the names of folders to be renamed

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-NewName<String>

The name the new folder(s) should have.

Required?	true
Default Value	
Accept Pipeline Input?	false

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.

Accept Pipeline Input?	false

Input Type

Depends on parameter Parameters can be piped by property name.

Return Values

None or Citrix.Broker.Admin.SDK.AdminFolder

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.AdminFolder object.

Examples

----- **EXAMPLE 1** -----

Rename-BrokerAdminFolder F1\XXX\YYY

Renames the folder called XXX within the folder F1\ to YYY

Rename-BrokerAppAssignmentPolicyRule

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
Reloads the access permissions and configuration service locations for the Broker Service.

Syntax

Reset-BrokerServiceGroupMembership -ConfigServiceInstance <PSObject[]> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Detailed Description

Enables you to reload Broker Service access permissions and configuration service locations. The Reset-BrokerServiceGroupMembership command must be run on at least one instance of the service type (Broker) after installation and registration with the configuration service. Without this operation, the Broker services will be unable to communicate with other services in the XenDesktop deployment. When the command is run, the services are updated when additional services are added to the deployment, provided that the configuration service is not stopped. The Reset-BrokerServiceGroupMembership command can be run again to refresh this information if automatic updates do not occur when new services are added to the deployment. If more than one configuration service instance is passed to the command, the first instance that meets the expected service type requirements is used.

Related topics

[Get-BrokerServiceInstance](#)

[Get-BrokerServiceStatus](#)

Parameters

-ConfigServiceInstance<PSObject[]>

Specifies the configuration service instance object that represents the service instance for the type 'InterService' that references a configuration service for the deployment.

Required?	true
Default Value	LocalHost
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

`Citrix.Broker.Sdk.ServiceInstance[]` Service instances containing a `ServiceInstance` object that refers to the central configuration service interservice interface can be piped to the `Reset-BrokerServiceGroupMembership` command.

Return Values

`Citrix.Broker.Sdk.ServiceInstance`

`Reset-BrokerServiceGroupMembership` returns opaque objects containing Configuration Service instances that are used by the Broker Service instance.

Notes

If the command fails, the following errors can be returned.

Error Codes

NoSuitableServiceInstance

None of the supplied service instance objects were suitable for resetting service group membership.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config | Reset-BrokerServiceGroupMembership
```

Reset the service group membership for a service in a deployment where the configuration service is configured and running on the same machine as the service.

----- **EXAMPLE 2** -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config -AdminAddress OtherServer.example.com | Reset-BrokerServiceGroupmembership
```

Reset the service group membership for a service in a deployment where the configuration service that is configured and running on a machine named 'OtherServer.example.com'.

Send-BrokerSessionMessage

May 04, 2016

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

May 04, 2016
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, allow access to any authenticated user, any user (whether authenticated or not), or only non-authenticated users. For a detailed description of this property see "help about_Broker_AccessPolicy".

Valid values are Filtered, AnyAuthenticated, Any, AnonymousOnly and FilteredOrAnonymous.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AllowRestart<Boolean>

Changes whether the user can restart sessions delivered from the rule's desktop group. Session restart is handled as follows: For sessions on single-session power-managed machines, the machine is powered off, and a new session launch request made; for sessions on multi-session machines, a logoff request is issued to the session, and a new session launch request made; otherwise the property is ignored.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Description<String>

Changes the description of the rule. The text is purely informational for the administrator; it is never visible to the end user.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Enabled<Boolean>

Changes whether the rule is enabled or disabled. A disabled rule is ignored when evaluating the site's access policy.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ExcludedClientIPFilterEnabled<Boolean>

Changes whether the excluded client IP address filter is enabled or disabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ExcludedClientIPs<IPAddressRange[]>

Changes the IP addresses of user devices explicitly denied access to the rule's desktop group. Addresses can be specified as simple numeric addresses or as subnet masks (for example, 10.40.37.5 or 10.40.0.0/16). This property forms part of the excluded client IP address filter.

Required?	false
Default Value	

Accept Pipeline Input?	false

-ExcludedClientNameFilterEnabled<Boolean>

Changes whether the excluded client names filter is enabled or disabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ExcludedClientNames<String[]>

Changes which names of user devices are explicitly denied access to the rule's desktop group. This property forms part of the excluded client names filter.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ExcludedSmartAccessFilterEnabled<Boolean>

Changes whether the excluded SmartAccess tags filter is enabled or disabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ExcludedSmartAccessTags<String[]>

Changes which SmartAccess tags explicitly deny access to the rule's desktop group if any occur in those provided by Access Gateway with the user's connection. This property forms part of the excluded SmartAccess tags filter.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ExcludedUserFilterEnabled<Boolean>

Changes whether the excluded users filter is enabled or disabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

Required?	false
Default Value	

Accept Pipeline Input?	false

-ExcludedUsers<User[]>

Changes which users and groups are explicitly denied access to the rule's desktop group. This property forms part of the excluded users filter.

Required?	false
Default Value	
Accept Pipeline Input?	false

-HdxSslEnabled<Boolean>

Indicates whether SSL encryption is enabled for sessions delivered from the rule's desktop group.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedClientIPFilterEnabled<Boolean>

Changes whether the included client IP address filter is enabled or disabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedClientIPs<IPAddressRange[]>

Changes which IP addresses of user devices allowed access to the rule's desktop group. Addresses can be specified as simple numeric addresses or as subnet masks (for example, 10.40.37.5 or 10.40.0.0/16). This property forms part of the included client IP address filter.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedClientNameFilterEnabled<Boolean>

Changes whether the included client name filter is enabled or disabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

Required?	false
Default Value	

Accept Pipeline Input?	false

-IncludedClientNames<String[]>

Changes which names of user devices are allowed access to the rule's desktop group. This property forms part of the included client names filter.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedSmartAccessFilterEnabled<Boolean>

Changes whether the included SmartAccess tags filter is enabled or disabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedSmartAccessTags<String[]>

Changes which SmartAccess tags grant access to the rule's desktop group if any occur in those provided by Access Gateway with the user's connection. This property forms part of the excluded SmartAccess tags filter.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedUserFilterEnabled<Boolean>

Changes whether the included users filter is enabled or disabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedUsers<User[]>

Changes which users and groups are granted access to the rule's desktop group. This property forms part of the included users filter.

Required?	false
Default Value	

Accept Pipeline Input?	false

-RemoveExcludedClientIPs<IPAddressRange[]>

Removes the specified user device IP addresses from the excluded client IP address filter of the rule.

See the ExcludedClientIPs parameter for more information.

Required?	false
Default Value	
Accept Pipeline Input?	false

-RemoveExcludedClientNames<String[]>

Removes the specified user device names from the excluded client names filter of the rule.

See the ExcludedClientNames parameter for more information.

Required?	false
Default Value	
Accept Pipeline Input?	false

-RemoveExcludedSmartAccessTags<String[]>

Removes the specified SmartAccess tags from the excluded SmartAccess tags filter of the rule.

See the ExcludedSmartAccessTags parameter for more information.

Required?	false
Default Value	
Accept Pipeline Input?	false

-RemoveExcludedUsers<User[]>

Removes the specified users and groups from the excluded users filter of the rule.

See the ExcludedUsers parameter for more information.

Required?	false
Default Value	
Accept Pipeline Input?	false

-RemoveIncludedClientIPs<IPAddressRange[]>

Removes the specified user device IP addresses from the included client IP address filter of the rule.

See the IncludedClientIPs parameter for more information.

Required?	false
Default Value	
Accept Pipeline Input?	false

-RemoveIncludedClientNames<String[]>

Removes the specified client names from the included client names filter of the rule.

See the IncludedClientNames parameter for more information.

Required?	false
Default Value	
Accept Pipeline Input?	false

-RemoveIncludedSmartAccessTags<String[]>

Removes the specified SmartAccess tags from the included SmartAccess tags filter of the rule.

See the IncludedSmartAccessTags parameter for more information.

Required?	false
Default Value	
Accept Pipeline Input?	false

-RemoveIncludedUsers<User[]>

Removes the specified users and groups from the included users filter of the rule.

See the IncludedUsers parameter for more information.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	

Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.AccessPolicyRule The access policy rule to be modified.

Return Values

None, or Citrix.Broker.Admin.SDK.AccessPolicyRule

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.AccessPolicyRule object.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Set-BrokerAccessPolicyRule 'Temp Staff' -AddIncludedUsers office\contractors
```

Adds user group OFFICE\contractors to the Temp Staff access policy rule. The resources that the group can access are dependent on the existing properties of the rule in addition to the site's assignment and entitlement policies.

----- **EXAMPLE 2** -----

```
C:\PS> Set-BrokerAccessPolicyRule 'Temp Staff' -ExcludedClientIPFilterEnabled $true -AddExcludedClientIPs '10.15.0.0/16' -AllowedConnections ViaAG
```

Modifies the Temp Staff access policy rule to remove access to any user device with an IP address matching 10.15.0.0/16, and requires that all connections by the rule must come through Access Gateway (assuming that the included SmartAccess tags filter is enabled).

Set-BrokerAccessPolicyRuleMetadata

May 04, 2016

Creates/Updates metadata key-value pairs for AccessPolicyRule

Syntax

```
Set-BrokerAccessPolicyRuleMetadata [-AccessPolicyRuleId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAccessPolicyRuleMetadata [-AccessPolicyRuleId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAccessPolicyRuleMetadata [-AccessPolicyRuleId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAccessPolicyRuleMetadata [-InputObject] <AccessPolicyRule[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAccessPolicyRuleMetadata [-InputObject] <AccessPolicyRule[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAccessPolicyRuleMetadata [-AccessPolicyRuleName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAccessPolicyRuleMetadata [-AccessPolicyRuleName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerAccessPolicyRuleMetadata cmdlet creates/updates metadata key-value pairs for AccessPolicyRule. The AccessPolicyRule can be specified by InputObject or piping.

Related topics

Parameters

-AccessPolicyRuleId<Int32>

Specifies the AccessPolicyRule object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<AccessPolicyRule[]>

Specifies the AccessPolicyRule objects whose Metadata is to be created/updated.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-AccessPolicyRuleName<String>

Specifies the AccessPolicyRule object whose Metadata is to be created/updated by name.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerAccessPolicyRule You can pipe the AccessPolicyRule to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerAccessPolicyRule

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerAccessPolicyRule object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerAccessPolicyRuleMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the AccessPolicyRule whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerAccessPolicyRule | Set-BrokerAccessPolicyRuleMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the AccessPolicyRule in the site

----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerAccessPolicyRuleMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the AccessPolicyRule in the site whose name is 'objname'

Set-BrokerAdminFolderMetadata

May 04, 2016

Creates/Updates metadata key-value pairs for AdminFolder

Syntax

```
Set-BrokerAdminFolderMetadata [-AdminFolderId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAdminFolderMetadata [-AdminFolderId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAdminFolderMetadata [-AdminFolderId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAdminFolderMetadata [-InputObject] <AdminFolder[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAdminFolderMetadata [-InputObject] <AdminFolder[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAdminFolderMetadata [-AdminFolderName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAdminFolderMetadata [-AdminFolderName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerAdminFolderMetadata cmdlet creates/updates metadata key-value pairs for AdminFolder. The AdminFolder can be specified by InputObject or piping.

Related topics

Parameters

-AdminFolderId<Int32>

Specifies the AdminFolder object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<AdminFolder[]>

Specifies the AdminFolder objects whose Metadata is to be created/updated.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-AdminFolderName<String>

Specifies the AdminFolder object whose Metadata is to be created/updated by name.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerAdminFolder You can pipe the AdminFolder to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerAdminFolder

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerAdminFolder object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerAdminFolderMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the AdminFolder whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerAdminFolder | Set-BrokerAdminFolderMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the AdminFolder in the site

----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerAdminFolderMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the AdminFolder in the site whose name is 'objname'

Set-BrokerAppAssignmentPolicyRule

May 04, 2016

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

May 04, 2016

Modifies an existing application rule in the site's entitlement policy.

Syntax

```
Set-BrokerAppEntitlementPolicyRule [-InputObject] <AppEntitlementPolicyRule[]> [-PassThru] [-AddExcludedUsers <User[]>] [-AddIncludedUsers <User[]>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-RemoveExcludedUsers <User[]>] [-RemoveIncludedUsers <User[]>] [-SessionReconnection <SessionReconnection>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAppEntitlementPolicyRule [-Name] <String> [-PassThru] [-AddExcludedUsers <User[]>] [-AddIncludedUsers <User[]>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-RemoveExcludedUsers <User[]>] [-RemoveIncludedUsers <User[]>] [-SessionReconnection <SessionReconnection>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerAppEntitlementPolicyRule cmdlet modifies an existing application rule in the site's entitlement policy.

An application rule in the entitlement policy defines the users who are allowed per-session access to a machine to run one or more applications published from the rule's desktop group.

Changing a rule does not affect existing sessions launched using the rule, but if the change removes an entitlement to a machine that was previously granted, users may be unable to reconnect to a disconnected session on that machine.

Related topics

[New-BrokerAppEntitlementPolicyRule](#)

[Get-BrokerAppEntitlementPolicyRule](#)

[Rename-BrokerAppEntitlementPolicyRule](#)

[Remove-BrokerAppEntitlementPolicyRule](#)

Parameters

-InputObject <AppEntitlementPolicyRule[]>

The application rule in the entitlement policy to be modified.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

The name of the application rule in the entitlement policy to be modified.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-AddExcludedUsers<User[]>

Adds the specified users to the excluded users filter of the rule, that is, the users and groups who are explicitly denied entitlements to run applications published from the desktop group.

See the ExcludedUsers parameter for more information.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AddIncludedUsers<User[]>

Adds the specified users to the included users filter of the rule, that is, the users and groups who are granted an entitlement to an application session by the rule.

See the IncludedUsers parameter for more information.

Required?	false

Default Value	
Accept Pipeline Input?	false

-Description<String>

Changes the description of the application rule. The text is purely informational for the administrator, it is never visible to the end user.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Enabled<Boolean>

Enables or disables the application rule. A disabled rule is ignored when evaluating the site's entitlement policy.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ExcludedUserFilterEnabled<Boolean>

Enables or disables the excluded users filter. If the filter is disabled then any user entries in the filter are ignored when entitlement policy rules are evaluated.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ExcludedUsers<User[]>

Changes the excluded users filter of the rule, that is, the users and groups who are explicitly denied entitlements to run applications published from the desktop group.

This can be used to exclude users or groups or users who would otherwise gain access by groups specified in the included

users filter.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedUserFilterEnabled<Boolean>

Enables or disables the included users filter. If the filter is disabled then any user who satisfies the requirements of the access policy is implicitly granted an entitlement to an application session by the application rule.

Users who would be implicitly granted access when the filter is disabled can still be explicitly denied access using the excluded users filter.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedUsers<User[]>

Changes the included users filter of the rule, that is, the users and groups who are granted an entitlement to an application session by the rule.

If a user appears explicitly in the excluded users filter of the rule or is a member of a group that appears in the excluded users filter, no entitlement is granted whether or not the user appears in the included users filter.

Required?	false
Default Value	
Accept Pipeline Input?	false

-RemoveExcludedUsers<User[]>

Removes the specified users from the excluded users filter of the application rule, that is, the users and groups who are explicitly denied entitlements to run applications published from the desktop group.

See the ExcludedUsers parameter for more information.

Required?	false

Default Value	
Accept Pipeline Input?	false

-RemoveIncludedUsers<User[]>

Removes the specified users from the included users filter of the rule, that is, the users and groups who are granted an entitlement to an application session by the rule.

See the IncludedUsers parameter for more information.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionReconnection<SessionReconnection>

Defines reconnection (roaming) behavior for sessions launched using this rule. Session reconnection control is an experimental and unsupported feature.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.AppEntitlementPolicyRule The application rule in the entitlement policy rule to be modified.

Return Values

None or Citrix.Broker.Admin.SDK.AppEntitlementPolicyRule

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.AppEntitlementPolicyRule object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerAppEntitlementPolicyRule 'Temp Workers' -AddIncludedUsers office\contractors
```

Adds the user group OFFICE\contractors to those entitled to run applications from the rule's associated desktop group. This grants all members of that group an entitlement to an application session from that group.

----- EXAMPLE 2 -----

```
C:\PS> Set-BrokerAppEntitlementPolicyRule 'Temp Workers' -Enabled $false
```

Disables the Temp Workers application rule in the entitlement policy. This prevents further application sessions being launched using this rule until it is re-enabled. However, access to existing application sessions is not affected.

Set-BrokerApplication

May 04, 2016

Changes the settings of an application to the value specified in the command.

Syntax

```
Set-BrokerApplication [-InputObject] <Application[]> [-PassThru] [-BrowserName <String>] [-ClientFolder <String>] [-CommandLineArguments <String>] [-CommandLineExecutable <String>] [-CpuPriorityLevel <CpuPriorityLevel>] [-Description <String>] [-Enabled <Boolean>] [-IconFromClient <Boolean>] [-IconUid <Int32>] [-PublishedName <String>] [-SecureCmdLineArgumentsEnabled <Boolean>] [-ShortcutAddedToDesktop <Boolean>] [-ShortcutAddedToStartMenu <Boolean>] [-StartMenuFolder <String>] [-UserFilterEnabled <Boolean>] [-Visible <Boolean>] [-WaitForPrinterCreation <Boolean>] [-WorkingDirectory <String>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplication [-Name] <String> [-PassThru] [-BrowserName <String>] [-ClientFolder <String>] [-CommandLineArguments <String>] [-CommandLineExecutable <String>] [-CpuPriorityLevel <CpuPriorityLevel>] [-Description <String>] [-Enabled <Boolean>] [-IconFromClient <Boolean>] [-IconUid <Int32>] [-PublishedName <String>] [-SecureCmdLineArgumentsEnabled <Boolean>] [-ShortcutAddedToDesktop <Boolean>] [-ShortcutAddedToStartMenu <Boolean>] [-StartMenuFolder <String>] [-UserFilterEnabled <Boolean>] [-Visible <Boolean>] [-WaitForPrinterCreation <Boolean>] [-WorkingDirectory <String>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerApplication cmdlet changes the value of one or more properties of an application, such as its CpuPriorityLevel or its CommandLineArguments, to the value specified in the command.

This cmdlet lets you change only the settings of the Application object, and not the relationships to other objects. For instance, it does not let you change which users can access this application, or change which desktop groups this application is published to. To do this, you need to remove the existing association, and then add a new association. The following example shows how to change the desktop group that an application is associated with from \$group1 to \$group2:

```
Remove-BrokerApplication -DesktopGroup $group1
```

```
Add-RemoveApplication -DesktopGroup $group2
```

You can change properties of both HostedOnDesktop and InstalledOnClient applications but it is not possible to change the ApplicationType. Also, the Name cannot be changed using this cmdlet; to do this, use the Rename-BrokerApplication cmdlet.

Related topics

[New-BrokerApplication](#)

[Add-BrokerApplication](#)

[Get-BrokerApplication](#)

[Remove-BrokerApplication](#)

[Rename-BrokerApplication](#)

Move-BrokerApplication

Parameters

-InputObject<Application[]>

Specifies the application to modify. The Uid of the application can also be substituted for the object.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the application to be modified.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-BrowserName<String>

Specifies the name of the application to modify. Note that the BrowserName cannot be changed in this manner; to modify the BrowserName of an application, use the Rename-BrokerApplication cmdlet.

Required?	false
Default Value	

Accept Pipeline Input?	false

-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

May 04, 2016

Creates/Updates metadata key-value pairs for ApplicationInstance

Syntax

```
Set-BrokerApplicationInstanceMetadata [-ApplicationInstanceId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationInstanceMetadata [-ApplicationInstanceId] <Int64> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationInstanceMetadata [-ApplicationInstanceId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationInstanceMetadata [-InputObject] <ApplicationInstance[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationInstanceMetadata [-InputObject] <ApplicationInstance[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerApplicationInstanceMetadata cmdlet creates/updates metadata key-value pairs for ApplicationInstance. The ApplicationInstance can be specified by InputObject or piping.

Related topics

Parameters

-ApplicationInstanceId<Int64>

Specifies the ApplicationInstance object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-InputObject<ApplicationInstance[]>

Specifies the ApplicationInstance objects whose Metadata is to be created/updated.

Required?	true
Default Value	

Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False

Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerApplicationInstance You can pipe the ApplicationInstance to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerApplicationInstance

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerApplicationInstance object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerApplicationInstanceMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the ApplicationInstance whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerApplicationInstance | Set-BrokerApplicationInstanceMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the ApplicationInstance in the site

Set-BrokerApplicationMetadata

May 04, 2016

Creates/Updates metadata key-value pairs for Application

Syntax

```
Set-BrokerApplicationMetadata [-ApplicationId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationMetadata [-ApplicationId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationMetadata [-ApplicationId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationMetadata [-InputObject] <Application[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationMetadata [-InputObject] <Application[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationMetadata [-ApplicationName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationMetadata [-ApplicationName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerApplicationMetadata cmdlet creates/updates metadata key-value pairs for Application. The Application can be specified by InputObject or piping.

Related topics

Parameters

-ApplicationId<Int32>

Specifies the Application object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-InputObject<Application[]>

Specifies the Application objects whose Metadata is to be created/updated.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-ApplicationName<String>

Specifies the Application object whose Metadata is to be created/updated by name.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerApplication You can pipe the Application to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerApplication

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerApplication object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerApplicationMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the Application whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerApplication | Set-BrokerApplicationMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the Application in the site

----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerApplicationMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the Application in the site whose name is 'objname'

Set-BrokerAssignmentPolicyRule

May 04, 2016

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

May 04, 2016

Creates/Updates metadata key-value pairs for AssignmentPolicyRule

Syntax

```
Set-BrokerAssignmentPolicyRuleMetadata [-AssignmentPolicyRuleId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAssignmentPolicyRuleMetadata [-AssignmentPolicyRuleId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAssignmentPolicyRuleMetadata [-AssignmentPolicyRuleId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAssignmentPolicyRuleMetadata [-InputObject] <AssignmentPolicyRule[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAssignmentPolicyRuleMetadata [-InputObject] <AssignmentPolicyRule[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAssignmentPolicyRuleMetadata [-AssignmentPolicyRuleName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAssignmentPolicyRuleMetadata [-AssignmentPolicyRuleName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerAssignmentPolicyRuleMetadata cmdlet creates/updates metadata key-value pairs for AssignmentPolicyRule. The AssignmentPolicyRule can be specified by InputObject or piping.

Related topics

Parameters

-AssignmentPolicyRuleId<Int32>

Specifies the AssignmentPolicyRule object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-InputObject<AssignmentPolicyRule[]>

Specifies the AssignmentPolicyRule objects whose Metadata is to be created/updated.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-AssignmentPolicyRuleName<String>

Specifies the AssignmentPolicyRule object whose Metadata is to be created/updated by name.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true

Default Value	
Accept Pipeline Input?	true (ByValue)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerAssignmentPolicyRule You can pipe the AssignmentPolicyRule to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerAssignmentPolicyRule

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerAssignmentPolicyRule object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerAssignmentPolicyRuleMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the AssignmentPolicyRule whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerAssignmentPolicyRule | Set-BrokerAssignmentPolicyRuleMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the AssignmentPolicyRule in the site

----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerAssignmentPolicyRuleMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the AssignmentPolicyRule in the site whose name is 'objname'

Set-BrokerCatalog

May 04, 2016

Sets the properties of a catalog.

Syntax

```
Set-BrokerCatalog [-InputObject] <Catalog[]> [-PassThru] [-Description <String>] [-IsRemotePC <Boolean>] [-MinimumFunctionalLevel <FunctionalLevel>] [-ProvisioningSchemeld <Guid>] [-PvsAddress <String>] [-PvsDomain <String>] [-RemotePCHypervisorConnectionUid <Int32>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerCatalog [-Name] <String> [-PassThru] [-Description <String>] [-IsRemotePC <Boolean>] [-MinimumFunctionalLevel <FunctionalLevel>] [-ProvisioningSchemeld <Guid>] [-PvsAddress <String>] [-PvsDomain <String>] [-RemotePCHypervisorConnectionUid <Int32>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerCatalog cmdlet sets properties of a catalog or set of catalogs. The catalog can be specified by name, in which case only one catalog can be specified, or one or more catalog instances can be passed to the command either by piping or by using the -InputObject parameter.

Related topics

[Get-BrokerCatalog](#)

[New-BrokerCatalog](#)

[Remove-BrokerCatalog](#)

[Rename-BrokerCatalog](#)

Parameters

-InputObject<Catalog[]>

Specifies the catalog objects to modify.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Identifies the catalog to modify.

Required?	true

Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-Description<String>

Supplies the new value of the Description property.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IsRemotePC<Boolean>

Supplies a new value for IsRemotePC.

IsRemotePC can only be enabled when:

- o SessionSupport is SingleSession
- o MachinesArePhysical is true.

IsRemotePC can only be set from true to false when no RemotePCAccount references this catalog, and when no Remote PC relationship exists between this catalog and a desktop group.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MinimumFunctionalLevel<FunctionalLevel>

The new minimum FunctionalLevel required for machines to work successfully in the catalog. If this is higher than the FunctionalLevel of any machines already in the catalog, they will immediately cease to function.

Valid values are L5, L7, L7_6

Required?	false
Default Value	
Accept Pipeline Input?	false

-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

May 04, 2016

Creates/Updates metadata key-value pairs for Catalog

Syntax

```
Set-BrokerCatalogMetadata [-CatalogId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

```
Set-BrokerCatalogMetadata [-CatalogId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

```
Set-BrokerCatalogMetadata [-CatalogId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

```
Set-BrokerCatalogMetadata [-InputObject] <Catalog[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

```
Set-BrokerCatalogMetadata [-InputObject] <Catalog[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

```
Set-BrokerCatalogMetadata [-CatalogName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

```
Set-BrokerCatalogMetadata [-CatalogName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

Detailed Description

The Set-BrokerCatalogMetadata cmdlet creates/updates metadata key-value pairs for Catalog. The Catalog can be specified by InputObject or piping.

Related topics

Parameters

-CatalogId<Int32>

Specifies the Catalog object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Catalog[]>

Specifies the Catalog objects whose Metadata is to be created/updated.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-CatalogName<String>

Specifies the Catalog object whose Metadata is to be created/updated by name.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerCatalog You can pipe the Catalog to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerCatalog

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerCatalog object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerCatalogMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the Catalog whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerCatalog | Set-BrokerCatalogMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the Catalog in the site

----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerCatalogMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the Catalog in the site whose name is 'objname'

Set-BrokerConfigurationSlotMetadata

May 04, 2016

Creates/Updates metadata key-value pairs for ConfigurationSlot

Syntax

```
Set-BrokerConfigurationSlotMetadata [-ConfigurationSlotId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerConfigurationSlotMetadata [-ConfigurationSlotId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerConfigurationSlotMetadata [-ConfigurationSlotId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerConfigurationSlotMetadata [-InputObject] <ConfigurationSlot[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerConfigurationSlotMetadata [-InputObject] <ConfigurationSlot[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerConfigurationSlotMetadata [-ConfigurationSlotName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerConfigurationSlotMetadata [-ConfigurationSlotName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerConfigurationSlotMetadata cmdlet creates/updates metadata key-value pairs for ConfigurationSlot. The ConfigurationSlot can be specified by InputObject or piping.

Related topics

Parameters

-ConfigurationSlotId<Int32>

Specifies the ConfigurationSlot object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-InputObject<ConfigurationSlot[]>

Specifies the ConfigurationSlot objects whose Metadata is to be created/updated.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-ConfigurationSlotName<String>

Specifies the ConfigurationSlot object whose Metadata is to be created/updated by name.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerConfigurationSlot You can pipe the ConfigurationSlot to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerConfigurationSlot

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerConfigurationSlot object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerConfigurationSlotMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the ConfigurationSlot whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerConfigurationSlot | Set-BrokerConfigurationSlotMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the ConfigurationSlot in the site

----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerConfigurationSlotMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the ConfigurationSlot in the site whose name is 'objname'

Set-BrokerControllerMetadata

May 04, 2016

Creates/Updates metadata key-value pairs for Controller

Syntax

```
Set-BrokerControllerMetadata [-ControllerId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerControllerMetadata [-ControllerId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerControllerMetadata [-ControllerId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerControllerMetadata [-InputObject] <Controller[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerControllerMetadata [-InputObject] <Controller[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerControllerMetadata [-ControllerName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerControllerMetadata [-ControllerName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerControllerMetadata cmdlet creates/updates metadata key-value pairs for Controller. The Controller can be specified by InputObject or piping.

Related topics

Parameters

-ControllerId<Int32>

Specifies the Controller object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-InputObject<Controller[]>

Specifies the Controller objects whose Metadata is to be created/updated.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-ControllerName<String>

Specifies the Controller object whose Metadata is to be created/updated by name.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerController You can pipe the Controller to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerController

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerController object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerControllerMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the Controller whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerController | Set-BrokerControllerMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the Controller in the site

----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerControllerMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the Controller in the site whose name is 'objname'

Set-BrokerDBConnection

May 04, 2016

Configures a database connection for the Broker Service.

Syntax

```
Set-BrokerDBConnection [-DBConnection] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [[-Force]]  
[<CommonParameters>]
```

Detailed Description

Specifies the database connection string for use by the currently selected Citrix Broker Service instance.

The service records the connection string and attempts to contact the specified database. If the database cannot initially be contacted the service retries the connection at intervals until contact with the database is successfully established.

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 the one on the local machine, or the one most recently specified using the -AdminAddress parameter of a Broker SDK cmdlet.

Related topics

[Get-BrokerServiceStatus](#)

[Get-BrokerDBConnection](#)

[Test-BrokerDBConnection](#)

Parameters

-DBConnection<String>

Specifies the database connection string to be used by the Broker Service. Passing in \$null will clear any existing database connection configured.

Required?	true
Default Value	

Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

-Force<SwitchParameter>

If present, allows the local administrator to set the connection string to null when there are problems contacting the database or other services.

Required?	false
Default Value	false
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Set-BrokerDBConnection cmdlet returns an object describing the status of the Broker Service together with extra diagnostics information. Possible values are:

-- OK:

The Broker Service instance is configured with a valid database and service schema. The service is operational.

-- DBUnconfigured:

No database connection string is set for the Broker Service instance.

-- DBRejectedConnection:

The database rejected the logon attempt from the Broker Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

-- InvalidDBConfigured:

The specified database does not exist, is not visible to the Broker Service instance, or the service's schema within the database is invalid.

-- DBNotFound:

The specified database could not be located with the configured connection string.

-- DBNewerVersionThanService:

The version of the Broker Service currently in use is newer than, and incompatible with, the version of the Broker Service schema on the database. Upgrade the Broker Service to a more recent version.

-- DBOlderVersionThanService:

The version of the Broker Service schema on the database is newer than, and incompatible with, the version of the Broker Service currently in use. Upgrade the database schema to a more recent version.

-- DBVersionChangeInProgress:

A database schema upgrade is in progress.

-- PendingFailure:

Connectivity between the Broker Service instance and the database has been lost. This may be a transitory network error, but may indicate a loss of connectivity that requires administrator intervention.

-- Failed:

Connectivity between the Broker Service instance and the database has been lost for an extended period of time, or has failed due to a configuration problem. The service instance cannot operate while its connection to the database is unavailable.

-- Unknown:

The status of the Broker Service cannot be determined.

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidDBConnectionString

The database connection string has an invalid format.

DatabaseConnectionDetailsAlreadyConfigured

There was already a database connection configured. After a configuration is set, it can only be set to \$null.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

```
C:\PS>Set-BrokerDBConnection "Server=dbserver\SQLEXPRESS;Database=XDDB;Trusted_Connection=True"
```

Configures the service instance to use a database called XDDB on an SQL Server Express database running on the machine called dbserver. Integrated Windows authentication is required.

----- EXAMPLE 2 -----

```
C:\PS>Set-BrokerDBConnection -DBConnection $null
```

Resets the service instance's database connection string. The service instance resets and returns to an idle state until a valid new database connection string is specified.

Set-BrokerDesktopGroup

May 04, 2016

Adjusts the settings of a broker desktop group.

Syntax

```
Set-BrokerDesktopGroup [-InputObject] <DesktopGroup[]> [-PassThru] [-AutomaticPowerOnForAssigned <Boolean>] [-AutomaticPowerOnForAssignedDuringPeak <Boolean>] [-ColorDepth <ColorDepth>] [-DeliveryType <DeliveryType>] [-Description <String>] [-Enabled <Boolean>] [-IconUid <Int32>] [-InMaintenanceMode <Boolean>] [-IsRemotePC <Boolean>] [-MinimumFunctionalLevel <FunctionalLevel>] [-OffPeakBufferSizePercent <Int32>] [-OffPeakDisconnectAction <SessionChangeHostingAction>] [-OffPeakDisconnectTimeout <Int32>] [-OffPeakExtendedDisconnectAction <SessionChangeHostingAction>] [-OffPeakExtendedDisconnectTimeout <Int32>] [-OffPeakLogOffAction <SessionChangeHostingAction>] [-OffPeakLogOffTimeout <Int32>] [-PeakBufferSizePercent <Int32>] [-PeakDisconnectAction <SessionChangeHostingAction>] [-PeakDisconnectTimeout <Int32>] [-PeakExtendedDisconnectAction <SessionChangeHostingAction>] [-PeakExtendedDisconnectTimeout <Int32>] [-PeakLogOffAction <SessionChangeHostingAction>] [-PeakLogOffTimeout <Int32>] [-ProtocolPriority <String[]>] [-PublishedName <String>] [-SecureIcaRequired <Boolean>] [-SettlementPeriodBeforeAutoShutdown <TimeSpan>] [-ShutdownDesktopsAfterUse <Boolean>] [-TimeZone <String>] [-TurnOnAddedMachine <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerDesktopGroup [-Name] <String> [-PassThru] [-AutomaticPowerOnForAssigned <Boolean>] [-AutomaticPowerOnForAssignedDuringPeak <Boolean>] [-ColorDepth <ColorDepth>] [-DeliveryType <DeliveryType>] [-Description <String>] [-Enabled <Boolean>] [-IconUid <Int32>] [-InMaintenanceMode <Boolean>] [-IsRemotePC <Boolean>] [-MinimumFunctionalLevel <FunctionalLevel>] [-OffPeakBufferSizePercent <Int32>] [-OffPeakDisconnectAction <SessionChangeHostingAction>] [-OffPeakDisconnectTimeout <Int32>] [-OffPeakExtendedDisconnectAction <SessionChangeHostingAction>] [-OffPeakExtendedDisconnectTimeout <Int32>] [-OffPeakLogOffAction <SessionChangeHostingAction>] [-OffPeakLogOffTimeout <Int32>] [-PeakBufferSizePercent <Int32>] [-PeakDisconnectAction <SessionChangeHostingAction>] [-PeakDisconnectTimeout <Int32>] [-PeakExtendedDisconnectAction <SessionChangeHostingAction>] [-PeakExtendedDisconnectTimeout <Int32>] [-PeakLogOffAction <SessionChangeHostingAction>] [-PeakLogOffTimeout <Int32>] [-ProtocolPriority <String[]>] [-PublishedName <String>] [-SecureIcaRequired <Boolean>] [-SettlementPeriodBeforeAutoShutdown <TimeSpan>] [-ShutdownDesktopsAfterUse <Boolean>] [-TimeZone <String>] [-TurnOnAddedMachine <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerDesktopGroup cmdlet is used to disable or enable an existing broker desktop group or to alter its settings.

Related topics

[Get-BrokerDesktopGroup](#)

[New-BrokerDesktopGroup](#)

[Rename-BrokerDesktopGroup](#)

[Remove-BrokerDesktopGroup](#)

Parameters

-InputObject<DesktopGroup[]>

Specifies the desktop groups to adjust.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the desktop groups to adjust, based on their Name property.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-AutomaticPowerOnForAssigned<Boolean>

Specifies whether assigned desktops in the desktop group should be automatically started at the start of peak time periods. Only relevant for groups whose DesktopKind is Private.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AutomaticPowerOnForAssignedDuringPeak<Boolean>

Specifies whether assigned desktops in the desktop group should be automatically started throughout peak time periods. Only relevant for groups whose DesktopKind is Private and which have AutomaticPowerOnForAssigned set to true.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ColorDepth<ColorDepth>

Specifies the color depth that the ICA session should use for desktops in this group. Valid values are FourBit, EightBit, SixteenBit, and TwentyFourBit.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DeliveryType<DeliveryType>

Specifies whether desktops, applications, or both, can be delivered from machines contained within the desktop group. Desktop groups with a DesktopKind of Private cannot be used to deliver both desktops and applications.

When changing the delivery type to desktops only, there must be no remaining desktop-hosted applications associated with the group, or application-specific assignment/entitlement policy rules for the group.

When changing the delivery type to applications only, there must be no remaining client-hosted applications associated with the group, or desktop-specific assignment/entitlement policy rules for the group.

Valid values are DesktopsOnly, AppsOnly, and DesktopsAndApps.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Description<String>

A description for this desktop group useful for administrators of the site.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Enabled<Boolean>

Whether the desktop group should be in the enabled state; disabled desktop groups do not appear to users.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IconUid<Int32>

The UID of the broker icon to be displayed to users for their desktop(s) in this desktop group.

Required?	false
Default Value	
Accept Pipeline Input?	false

-InMaintenanceMode<Boolean>

Whether the desktop should be put into maintenance mode; a desktop group in maintenance mode will not allow users to connect or reconnect to their desktops.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IsRemotePC<Boolean>

Supplies a new value for IsRemotePC.

IsRemotePC can only be enabled when:

- o SessionSupport is SingleSession

- o DeliveryType is DesktopsOnly

- o DesktopKind is Private

IsRemotePC can be switched from true to false only if no RemotePC relationship exists between a catalog and this desktop group.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MinimumFunctionalLevel<FunctionalLevel>

The new minimum FunctionalLevel required for machines to work successfully in the desktop group. If this is higher than the FunctionalLevel of any machines already in the desktop group, they will immediately cease to function.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OffPeakBufferSizePercent<Int32>

The percentage of machines in the desktop group that should be kept available in an idle state outside peak hours.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OffPeakDisconnectAction<SessionChangeHostingAction>

The action to be performed after a configurable period of a user session disconnecting outside peak hours. Possible values are Nothing, Suspend, or Shutdown.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OffPeakDisconnectTimeout<Int32>

The number of minutes before the configured action should be performed after a user session disconnects outside peak hours.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OffPeakExtendedDisconnectAction<SessionChangeHostingAction>

The action to be performed after a second configurable period of a user session disconnecting outside peak hours. Possible values are Nothing, Suspend, or Shutdown.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OffPeakExtendedDisconnectTimeout<Int32>

The number of minutes before the second configured action should be performed after a user session disconnects outside peak hours.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OffPeakLogOffAction<SessionChangeHostingAction>

The action to be performed after a configurable period of a user session ending outside peak hours. Possible values are Nothing, Suspend, or Shutdown.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OffPeakLogOffTimeout<Int32>

The number of minutes before the configured action should be performed after a user session ends outside peak hours.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PeakBufferSizePercent<Int32>

The percentage of machines in the desktop group that should be kept available in an idle state in peak hours.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PeakDisconnectAction<SessionChangeHostingAction>

The action to be performed after a configurable period of a user session disconnecting in peak hours. Possible values are Nothing, Suspend, or Shutdown.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PeakDisconnectTimeout<Int32>

The number of minutes before the configured action should be performed after a user session disconnects in peak hours.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PeakExtendedDisconnectAction<SessionChangeHostingAction>

The action to be performed after a second configurable period of a user session disconnecting in peak hours. Possible values are Nothing, Suspend, or Shutdown.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PeakExtendedDisconnectTimeout<Int32>

The number of minutes before the second configured action should be performed after a user session disconnects in peak hours.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PeakLogOffAction<SessionChangeHostingAction>

The action to be performed after a configurable period of a user session ending in peak hours. Possible values are Nothing, Suspend, or Shutdown.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PeakLogOffTimeout<Int32>

The number of minutes before the configured action should be performed after a user session ends in peak hours.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ProtocolPriority<String[]>

A list of protocol names in the order in which they should be attempted for use during connection.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PublishedName<String>

The name that will be displayed to users for their desktop(s) in this desktop group.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SecureIcaRequired<Boolean>

Whether HDX connections to desktops in the new desktop group require the use of a secure protocol.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SettlementPeriodBeforeAutoShutdown<TimeSpan>

Time after a session ends during which automatic shutdown requests (for example, shutdown after use, idle pool management) are deferred. Any outstanding shutdown request takes effect after the settlement period expires. This is typically used to configure time to allow logoff scripts to complete.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ShutdownDesktopsAfterUse<Boolean>

Whether desktops in this desktop group should be automatically shut down when each user session completes (only relevant to power-managed desktops).

Required?	false
Default Value	
Accept Pipeline Input?	false

-TimeZone<String>

The time zone in which this desktop group's machines reside.

The time zone must be specified for any of the group's automatic power management settings to take effect. Automatic power management operations include pool management (power time schemes), reboot schedules, session disconnect and logoff actions, and powering on assigned machines etc.

Required?	false
Default Value	
Accept Pipeline Input?	false

-TurnOnAddedMachine<Boolean>

This flag specifies whether the Broker Service should attempt to power on machines when they are added to the desktop group.

Required?	false
Default Value	

Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.DesktopGroup You can pipe the desktop groups to be adjusted to Set-BrokerDesktopGroup.

Return Values

None or Citrix.Broker.Admin.SDK.DesktopGroup

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.DesktopGroup object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerDesktopGroup EMEA* -InMaintenanceMode $true -PassThru
```

Sets all desktop groups with names starting "EMEA" into maintenance mode, returning the set of desktop groups.

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerDesktopGroup -InMaintenanceMode $true | Set-BrokerDesktopGroup -Enabled $false
```

Disable all desktop groups that are in maintenance mode.

Set-BrokerDesktopGroupMetadata

May 04, 2016

Creates/Updates metadata key-value pairs for DesktopGroup

Syntax

```
Set-BrokerDesktopGroupMetadata [-DesktopGroupId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerDesktopGroupMetadata [-DesktopGroupId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerDesktopGroupMetadata [-DesktopGroupId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerDesktopGroupMetadata [-InputObject] <DesktopGroup[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerDesktopGroupMetadata [-InputObject] <DesktopGroup[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerDesktopGroupMetadata [-DesktopGroupName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerDesktopGroupMetadata [-DesktopGroupName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerDesktopGroupMetadata cmdlet creates/updates metadata key-value pairs for DesktopGroup. The DesktopGroup can be specified by InputObject or piping.

Related topics

Parameters

-DesktopGroupId<Int32>

Specifies the DesktopGroup object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<DesktopGroup[]>

Specifies the DesktopGroup objects whose Metadata is to be created/updated.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-DesktopGroupName<String>

Specifies the DesktopGroup object whose Metadata is to be created/updated by name.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerDesktopGroup You can pipe the DesktopGroup to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerDesktopGroup

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerDesktopGroup object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerDesktopGroupMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the DesktopGroup whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerDesktopGroup | Set-BrokerDesktopGroupMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the DesktopGroup in the site

----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerDesktopGroupMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the DesktopGroup in the site whose name is 'objname'

Set-BrokerEntitlementPolicyRule

May 04, 2016

Modifies an existing desktop rule in the site's entitlement policy.

Syntax

```
Set-BrokerEntitlementPolicyRule [-InputObject] <EntitlementPolicyRule[]> [-PassThru] [-AddExcludedUsers <User[]>] [-AddIncludedUsers <User[]>] [-ColorDepth <ColorDepth>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-IconUid <Int32>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-PublishedName <String>] [-RemoveExcludedUsers <User[]>] [-RemoveIncludedUsers <User[]>] [-SecureIcaRequired <Boolean>] [-SessionReconnection <SessionReconnection>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerEntitlementPolicyRule [-Name] <String> [-PassThru] [-AddExcludedUsers <User[]>] [-AddIncludedUsers <User[]>] [-ColorDepth <ColorDepth>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-IconUid <Int32>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-PublishedName <String>] [-RemoveExcludedUsers <User[]>] [-RemoveIncludedUsers <User[]>] [-SecureIcaRequired <Boolean>] [-SessionReconnection <SessionReconnection>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerEntitlementPolicyRule cmdlet modifies an existing desktop rule in the site's entitlement policy.

A desktop rule in the entitlement policy defines the users who are allowed per-session access to a machine from the rule's associated desktop group to run a full desktop session.

Changing a rule does not affect existing sessions launched using the rule, but if the change removes an entitlement to a machine that was previously granted, users may be unable to reconnect to a disconnected session on that machine.

Related topics

[New-BrokerEntitlementPolicyRule](#)

[Get-BrokerEntitlementPolicyRule](#)

[Rename-BrokerEntitlementPolicyRule](#)

[Remove-BrokerEntitlementPolicyRule](#)

Parameters

-InputObject <EntitlementPolicyRule[]>

The desktop rule in the entitlement policy to be modified.

Required?	true
Default Value	

Accept Pipeline Input?	true (ByValue)

-Name<String>

The name of the desktop rule in the entitlement policy to be modified.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-AddExcludedUsers<User[]>

Adds the specified users to the excluded users filter of the rule, that is, the users and groups who are explicitly denied an entitlement to a desktop session from this rule.

See the ExcludedUsers parameter for more information.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AddIncludedUsers<User[]>

Adds the specified users to the included users filter of the rule, that is, the users and groups who are granted an entitlement to a desktop session by the rule.

See the IncludedUsers parameter for more information.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ColorDepth<ColorDepth>

Changes the color depth of any desktop sessions launched by a user from this entitlement. Existing sessions are not affected.

Valid values are \$null, FourBit, EightBit, SixteenBit, and TwentyFourBit.

A null value indicates that the equivalent setting from the rule's desktop group is used.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Description<String>

Changes the description of the desktop rule. The text may be visible to the end user, for example, as a tooltip associated with the desktop entitlement.

A null value indicates that the equivalent setting from the rule's desktop group is used.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Enabled<Boolean>

Enables or disables the desktop rule. A disabled rule is ignored when evaluating the site's entitlement policy.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ExcludedUserFilterEnabled<Boolean>

Enables or disables the excluded users filter. If the filter is disabled then any user entries in the filter are ignored when entitlement policy rules are evaluated.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ExcludedUsers<User[]>

Changes the excluded users filter of the desktop rule, that is, the users and groups who are explicitly denied an entitlement to a desktop session from this rule.

This can be used to exclude users or groups who would otherwise gain access by groups specified in the included users filter.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IconUid<Int32>

Changes the icon (identified by its unique ID) for the published desktop entitlement as seen by the user.

A null value indicates that the equivalent setting from the rule's desktop group is used.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedUserFilterEnabled<Boolean>

Enables or disables the included users filter. If the filter is disabled then any user who satisfies the requirements of the access policy is implicitly granted an entitlement to a desktop session by the rule.

Users who would be implicitly granted access when the filter is disabled can still be explicitly denied access using the excluded users filter.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IncludedUsers<User[]>

Changes the included users filter of the desktop rule, that is, the users and groups who are granted an entitlement to a desktop session by the rule.

If a user appears explicitly in the excluded users filter of the rule or is a member of a group that appears in the excluded users filter, no entitlement is granted whether or not the user appears in the included users filter.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PublishedName<String>

Changes the name of the published desktop entitlement as seen by the user.

A null value indicates that the equivalent setting from the rule's desktop group is used.

Required?	false
Default Value	
Accept Pipeline Input?	false

-RemoveExcludedUsers<User[]>

Removes the specified users from the excluded users filter of the rule, that is, the users and groups who are explicitly denied an entitlement to a desktop session from this rule.

See the ExcludedUsers parameter for more information.

Required?	false
Default Value	

Accept Pipeline Input?	false

-RemoveIncludedUsers<User[]>

Removes the specified users from the included users filter of the desktop rule, that is, the users and groups who are granted an entitlement to a desktop session by the rule.

See the IncludedUsers parameter for more information.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SecureIcaRequired<Boolean>

Changes whether the desktop rule requires the SecureICA protocol for desktop sessions launched using the entitlement.

A null value indicates that the equivalent setting from the rule's desktop group is used.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SessionReconnection<SessionReconnection>

Defines reconnection (roaming) behavior for sessions launched using this rule. Session reconnection control is an experimental and unsupported feature.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.EntitlementPolicyRule The desktop rule in the entitlement policy to be modified.

Return Values

None or Citrix.Broker.Admin.SDK.EntitlementPolicyRule

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.EntitlementPolicyRule object.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Set-BrokerEntitlementPolicyRule 'Temp Workers' -AddIncludedUsers office\contractors
```

Adds the user group OFFICE\contractors to the Temp Workers desktop rule of the entitlement policy. This grants all members of that group an entitlement to a desktop session in the rule's associated desktop group. The session properties of the desktops obtained using the rule are determined by the rule's other properties.

----- **EXAMPLE 2** -----

```
C:\PS> Set-BrokerEntitlementPolicyRule 'Temp Workers' -Enabled $false
```

Disables the Temp Workers desktop rule in the entitlement policy. This prevents further desktop sessions being launched using this rule until it is re-enabled. However, access to existing desktop sessions is not affected.

Set-BrokerEntitlementPolicyRuleMetadata

May 04, 2016

Creates/Updates metadata key-value pairs for EntitlementPolicyRule

Syntax

```
Set-BrokerEntitlementPolicyRuleMetadata [-EntitlementPolicyRuleId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerEntitlementPolicyRuleMetadata [-EntitlementPolicyRuleId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerEntitlementPolicyRuleMetadata [-EntitlementPolicyRuleId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerEntitlementPolicyRuleMetadata [-InputObject] <EntitlementPolicyRule[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerEntitlementPolicyRuleMetadata [-InputObject] <EntitlementPolicyRule[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerEntitlementPolicyRuleMetadata [-EntitlementPolicyRuleName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerEntitlementPolicyRuleMetadata [-EntitlementPolicyRuleName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerEntitlementPolicyRuleMetadata cmdlet creates/updates metadata key-value pairs for EntitlementPolicyRule. The EntitlementPolicyRule can be specified by InputObject or piping.

Related topics

Parameters

-EntitlementPolicyRuleId<Int32>

Specifies the EntitlementPolicyRule object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-InputObject<EntitlementPolicyRule[]>

Specifies the EntitlementPolicyRule objects whose Metadata is to be created/updated.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-EntitlementPolicyRuleName<String>

Specifies the EntitlementPolicyRule object whose Metadata is to be created/updated by name.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true

Default Value	
Accept Pipeline Input?	true (ByValue)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerEntitlementPolicyRule You can pipe the EntitlementPolicyRule to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerEntitlementPolicyRule

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerEntitlementPolicyRule object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerEntitlementPolicyRuleMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the EntitlementPolicyRule whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerEntitlementPolicyRule | Set-BrokerEntitlementPolicyRuleMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the EntitlementPolicyRule in the site

----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerEntitlementPolicyRuleMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the EntitlementPolicyRule in the site whose name is 'objname'

Set-BrokerHostingPowerAction

May 04, 2016

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

May 04, 2016

Creates/Updates metadata key-value pairs for HostingPowerAction

Syntax

```
Set-BrokerHostingPowerActionMetadata [-HostingPowerActionId] <Int64> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHostingPowerActionMetadata [-HostingPowerActionId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHostingPowerActionMetadata [-HostingPowerActionId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHostingPowerActionMetadata [-InputObject] <HostingPowerAction[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHostingPowerActionMetadata [-InputObject] <HostingPowerAction[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHostingPowerActionMetadata [-HostingPowerActionName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHostingPowerActionMetadata [-HostingPowerActionName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerHostingPowerActionMetadata cmdlet creates/updates metadata key-value pairs for HostingPowerAction. The HostingPowerAction can be specified by InputObject or piping.

Related topics

Parameters

-HostingPowerActionId<Int64>

Specifies the HostingPowerAction object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<HostingPowerAction[]>

Specifies the HostingPowerAction objects whose Metadata is to be created/updated.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-HostingPowerActionName<String>

Specifies the HostingPowerAction object whose Metadata is to be created/updated by name.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerHostingPowerAction You can pipe the HostingPowerAction to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerHostingPowerAction

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerHostingPowerAction object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerHostingPowerActionMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the HostingPowerAction whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerHostingPowerAction | Set-BrokerHostingPowerActionMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the HostingPowerAction in the site

----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerHostingPowerActionMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the HostingPowerAction in the site whose name is 'objname'

Set-BrokerHypervisorAlertMetadata

May 04, 2016

Creates/Updates metadata key-value pairs for HypervisorAlert

Syntax

```
Set-BrokerHypervisorAlertMetadata [-HypervisorAlertId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorAlertMetadata [-HypervisorAlertId] <Int64> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorAlertMetadata [-HypervisorAlertId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorAlertMetadata [-InputObject] <HypervisorAlert[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorAlertMetadata [-InputObject] <HypervisorAlert[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerHypervisorAlertMetadata cmdlet creates/updates metadata key-value pairs for HypervisorAlert. The HypervisorAlert can be specified by InputObject or piping.

Related topics

Parameters

-HypervisorAlertId<Int64>

Specifies the HypervisorAlert object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-InputObject<HypervisorAlert[]>

Specifies the HypervisorAlert objects whose Metadata is to be created/updated.

Required?	true
Default Value	

Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False

Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerHypervisorAlert You can pipe the HypervisorAlert to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerHypervisorAlert

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerHypervisorAlert object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerHypervisorAlertMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the HypervisorAlert whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerHypervisorAlert | Set-BrokerHypervisorAlertMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the HypervisorAlert in the site

Set-BrokerHypervisorConnection

May 04, 2016

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

May 04, 2016

Creates/Updates metadata key-value pairs for HypervisorConnection

Syntax

```
Set-BrokerHypervisorConnectionMetadata [-HypervisorConnectionId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorConnectionMetadata [-HypervisorConnectionId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorConnectionMetadata [-HypervisorConnectionId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorConnectionMetadata [-InputObject] <HypervisorConnection[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorConnectionMetadata [-InputObject] <HypervisorConnection[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorConnectionMetadata [-HypervisorConnectionName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorConnectionMetadata [-HypervisorConnectionName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerHypervisorConnectionMetadata cmdlet creates/updates metadata key-value pairs for HypervisorConnection. The HypervisorConnection can be specified by InputObject or piping.

Related topics

Parameters

-HypervisorConnectionId<Int32>

Specifies the HypervisorConnection object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<HypervisorConnection[]>

Specifies the HypervisorConnection objects whose Metadata is to be created/updated.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-HypervisorConnectionName<String>

Specifies the HypervisorConnection object whose Metadata is to be created/updated by name.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true

Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerHypervisorConnection You can pipe the HypervisorConnection to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerHypervisorConnection

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerHypervisorConnection object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerHypervisorConnectionMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the HypervisorConnection whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerHypervisorConnection | Set-BrokerHypervisorConnectionMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the HypervisorConnection in the site

----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerHypervisorConnectionMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the HypervisorConnection in the site whose name is 'objname'

Set-BrokerIconMetadata

May 04, 2016

Creates/Updates metadata key-value pairs for Icon

Syntax

```
Set-BrokerIconMetadata [-IconId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-BrokerIconMetadata [-IconId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Set-BrokerIconMetadata [-IconId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Set-BrokerIconMetadata [-InputObject] <Icon[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-  
AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerIconMetadata [-InputObject] <Icon[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

The Set-BrokerIconMetadata cmdlet creates/updates metadata key-value pairs for Icon. The Icon can be specified by InputObject or piping.

Related topics

Parameters

-IconId<Int32>

Specifies the Icon object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Icon[]>

Specifies the Icon objects whose Metadata is to be created/updated.

Required?	true
Default Value	

Accept Pipeline Input?	true (ByValue)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False

Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerIcon You can pipe the Icon to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerIcon

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerIcon object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerIconMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the Icon whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerIcon | Set-BrokerIconMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the Icon in the site

Set-BrokerLeaseMetadata

May 04, 2016

Creates/Updates metadata key-value pairs for Lease

Syntax

```
Set-BrokerLeaseMetadata [-LeaseId] <Int64> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerLeaseMetadata [-LeaseId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerLeaseMetadata [-LeaseId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerLeaseMetadata [-InputObject] <Lease[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerLeaseMetadata [-InputObject] <Lease[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerLeaseMetadata [-LeaseName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerLeaseMetadata [-LeaseName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerLeaseMetadata cmdlet creates/updates metadata key-value pairs for Lease. The Lease can be specified by InputObject or piping.

Related topics

Parameters

-LeaseId<Int64>

Specifies the Lease object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Lease[]>

Specifies the Lease objects whose Metadata is to be created/updated.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-LeaseName<String>

Specifies the Lease object whose Metadata is to be created/updated by name.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerLease You can pipe the Lease to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerLease

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerLease object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerLeaseMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the Lease whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerLease | Set-BrokerLeaseMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the Lease in the site

----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerLeaseMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the Lease in the site whose name is 'objname'

Set-BrokerMachine

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Creates/Updates metadata key-value pairs for MachineCommand

Syntax

```
Set-BrokerMachineCommandMetadata [-MachineCommandId] <Int64> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineCommandMetadata [-MachineCommandId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineCommandMetadata [-MachineCommandId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineCommandMetadata [-InputObject] <MachineCommand[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineCommandMetadata [-InputObject] <MachineCommand[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerMachineCommandMetadata cmdlet creates/updates metadata key-value pairs for MachineCommand. The MachineCommand can be specified by InputObject or piping.

Related topics

Parameters

-MachineCommandId<Int64>

Specifies the MachineCommand object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<MachineCommand[]>

Specifies the MachineCommand objects whose Metadata is to be created/updated.

Required?	true
Default Value	

Accept Pipeline Input?	true (ByValue)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False

Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerMachineCommand You can pipe the MachineCommand to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerMachineCommand

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerMachineCommand object.

Examples

----- **EXAMPLE 1** -----

C:\PS> Set-BrokerMachineCommandMetadata -InputObject \$obj-Uid -Name "MyMetadataName" -Value "1234"
This command creates/updates the Metadata "MyMetadataName" key-value pair for the MachineCommand whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerMachineCommand | Set-BrokerMachineCommandMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the MachineCommand in the site

Set-BrokerMachineConfiguration

May 04, 2016

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

May 04, 2016

Creates/Updates metadata key-value pairs for MachineConfiguration

Syntax

```
Set-BrokerMachineConfigurationMetadata [-MachineConfigurationId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineConfigurationMetadata [-MachineConfigurationId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineConfigurationMetadata [-MachineConfigurationId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineConfigurationMetadata [-InputObject] <MachineConfiguration[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineConfigurationMetadata [-InputObject] <MachineConfiguration[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineConfigurationMetadata [-MachineConfigurationName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineConfigurationMetadata [-MachineConfigurationName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerMachineConfigurationMetadata cmdlet creates/updates metadata key-value pairs for MachineConfiguration. The MachineConfiguration can be specified by InputObject or piping.

Related topics

Parameters

-MachineConfigurationId<Int32>

Specifies the MachineConfiguration object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-InputObject<MachineConfiguration[]>

Specifies the MachineConfiguration objects whose Metadata is to be created/updated.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-MachineConfigurationName<String>

Specifies the MachineConfiguration object whose Metadata is to be created/updated by name.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true

Default Value	
Accept Pipeline Input?	true (ByValue)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerMachineConfiguration You can pipe the MachineConfiguration to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerMachineConfiguration

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerMachineConfiguration object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerMachineConfigurationMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the MachineConfiguration whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerMachineConfiguration | Set-BrokerMachineConfigurationMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the MachineConfiguration in the site

----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerMachineConfigurationMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the MachineConfiguration in the site whose name is 'objname'

Set-BrokerMachineMaintenanceMode

May 04, 2016

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

May 04, 2016

Creates/Updates metadata key-value pairs for Machine

Syntax

```
Set-BrokerMachineMetadata [-MachineId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineMetadata [-MachineId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineMetadata [-MachineId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineMetadata [-InputObject] <Machine[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineMetadata [-InputObject] <Machine[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineMetadata [-MachineName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineMetadata [-MachineName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerMachineMetadata cmdlet creates/updates metadata key-value pairs for Machine. The Machine can be specified by InputObject or piping.

Related topics

Parameters

-MachineId<Int32>

Specifies the Machine object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-InputObject<Machine[]>

Specifies the Machine objects whose Metadata is to be created/updated.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-MachineName<String>

Specifies the Machine object whose Metadata is to be created/updated by name.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerMachine You can pipe the Machine to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerMachine

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerMachine object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerMachineMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the Machine whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerMachine | Set-BrokerMachineMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the Machine in the site

----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerMachineMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the Machine in the site whose name is 'objname'

Set-BrokerPowerTimeScheme

May 04, 2016
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

May 04, 2016

Creates/Updates metadata key-value pairs for PowerTimeScheme

Syntax

```
Set-BrokerPowerTimeSchemeMetadata [-PowerTimeSchemeId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerPowerTimeSchemeMetadata [-PowerTimeSchemeId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerPowerTimeSchemeMetadata [-PowerTimeSchemeId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerPowerTimeSchemeMetadata [-InputObject] <PowerTimeScheme[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerPowerTimeSchemeMetadata [-InputObject] <PowerTimeScheme[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerPowerTimeSchemeMetadata [-PowerTimeSchemeName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerPowerTimeSchemeMetadata [-PowerTimeSchemeName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerPowerTimeSchemeMetadata cmdlet creates/updates metadata key-value pairs for PowerTimeScheme. The PowerTimeScheme can be specified by InputObject or piping.

Related topics

Parameters

-PowerTimeSchemeId<Int32>

Specifies the PowerTimeScheme object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-InputObject<PowerTimeScheme[]>

Specifies the PowerTimeScheme objects whose Metadata is to be created/updated.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-PowerTimeSchemeName<String>

Specifies the PowerTimeScheme object whose Metadata is to be created/updated by name.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerPowerTimeScheme You can pipe the PowerTimeScheme to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerPowerTimeScheme

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerPowerTimeScheme object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerPowerTimeSchemeMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the PowerTimeScheme whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerPowerTimeScheme | Set-BrokerPowerTimeSchemeMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the PowerTimeScheme in the site

----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerPowerTimeSchemeMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the PowerTimeScheme in the site whose name is 'objname'

Set-BrokerPrivateDesktop

May 04, 2016

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

May 04, 2016

Creates/Updates metadata key-value pairs for RebootCycle

Syntax

```
Set-BrokerRebootCycleMetadata [-RebootCycleId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerRebootCycleMetadata [-RebootCycleId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerRebootCycleMetadata [-RebootCycleId] <Int64> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerRebootCycleMetadata [-InputObject] <RebootCycle[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerRebootCycleMetadata [-InputObject] <RebootCycle[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerRebootCycleMetadata cmdlet creates/updates metadata key-value pairs for RebootCycle. The RebootCycle can be specified by InputObject or piping.

Related topics

Parameters

-RebootCycleId<Int64>

Specifies the RebootCycle object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-InputObject<RebootCycle[]>

Specifies the RebootCycle objects whose Metadata is to be created/updated.

Required?	true
Default Value	

Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False

Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerRebootCycle You can pipe the RebootCycle to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerRebootCycle

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerRebootCycle object.

Examples

----- **EXAMPLE 1** -----

C:\PS> Set-BrokerRebootCycleMetadata -InputObject \$obj-Uid -Name "MyMetadataName" -Value "1234"
This command creates/updates the Metadata "MyMetadataName" key-value pair for the RebootCycle whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerRebootCycle | Set-BrokerRebootCycleMetadata -Name "MyMetadataName" -Value "1234"

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the RebootCycle in the site

Set-BrokerRebootSchedule

May 04, 2016
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

May 04, 2016

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

May 04, 2016

Sets properties of a session.

Syntax

```
Set-BrokerSession [-InputObject] <Session[]> [-PassThru] [-Hidden <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-BrokerSession [-SessionKey] <Guid> [-PassThru] [-Hidden <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

The Set-BrokerSession cmdlet sets properties on a session or set of sessions. You can specify a single session by Uid or SessionKey (Guid) or multiple session instances can be passed to the command by piping or using the -InputObject parameter.

Related topics

[Get-BrokerSession](#)

Parameters

-InputObject<Session[]>

The session instances whose properties you want to set.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-SessionKey<Guid>

The session key (Guid) of the session whose properties you want to set.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-Hidden<Boolean>

Changes whether the session is hidden or not. Hidden sessions are treated as though they do not exist when brokering sessions; a hidden session cannot be reconnected to, but a new session may be launched using the same entitlement.

A session may be hidden automatically by the system when an attempt is made to reconnect to a session that is unreachable due to, for example, the failure of a VM's hypervisor. This allows a new session to be brokered provided that other machines in the same desktop group are still available. If the original session still exists after the hypervisor is restored then it must be unhidden to allow the user to reconnect to it.

Only sessions for shared desktops or applications can be hidden or unhidden. Private desktop or application sessions can never be hidden.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host

name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.Session You can pipe in the sessions whose properties you want to set.

Return Values

None or Citrix.Broker.Admin.SDK.Session

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.Session object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> $sessions = Get-BrokerSession -User ACCOUNTS\JohnSmith -Hidden $true
```

```
C:\PS> $sessions | Set-BrokerSession -Hidden $false
```

Finds all sessions in the site for user John Smith that have been hidden, and makes them available for reconnection again.

Set-BrokerSessionLinger

May 04, 2016

Updates the values of one or more desktop group session linger settings.

Syntax

```
Set-BrokerSessionLinger [-InputObject] <SessionLinger[]> [-PassThru] [-Enabled <Boolean>] [-MaxAverageLoadThreshold <Int32>] [-MaxLoadPerMachineThreshold <Int32>] [-MaxTimeBeforeDisconnect <TimeSpan>] [-MaxTimeBeforeTerminate <TimeSpan>] [-UserFilterEnabled <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSessionLinger [-DesktopGroupName] <String> [-PassThru] [-Enabled <Boolean>] [-MaxAverageLoadThreshold <Int32>] [-MaxLoadPerMachineThreshold <Int32>] [-MaxTimeBeforeDisconnect <TimeSpan>] [-MaxTimeBeforeTerminate <TimeSpan>] [-UserFilterEnabled <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerSessionLinger cmdlet is used to alter the settings of an existing desktop group session linger setting.

Related topics

[New-BrokerSessionLinger](#)

[Get-BrokerSessionLinger](#)

[Remove-BrokerSessionLinger](#)

Parameters

-InputObject <SessionLinger[]>

The session linger to be modified.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-DesktopGroupName <String>

The name of the desktop group whose session linger setting is to be modified.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-Enabled<Boolean>

Boolean that indicates if the session linger setting is to be enabled or disabled.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MaxAverageLoadThreshold<Int32>

Specifies the average load threshold across the desktop group. When the threshold hits, lingering sessions across the group be terminated to reduce load. Sessions that have been lingering the longest will be chosen first.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MaxLoadPerMachineThreshold<Int32>

Specifies the maximum load threshold per machine in the desktop group. When the threshold hits, lingering sessions on each loaded machine will be terminated to reduce load. Sessions that have been lingering the longest will be chosen first.

Required?	false
Default Value	

Accept Pipeline Input?	false

-MaxTimeBeforeDisconnect<TimeSpan>

Specifies the time by which a lingering session will be disconnected. The disconnect time cannot be greater than the terminate timer (if enabled). When the disconnect and terminate times are the same, the terminate time takes precedence.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MaxTimeBeforeTerminate<TimeSpan>

Specifies the time by which a lingering session will be terminated. When the disconnect and terminate times are the same, the terminate time takes precedence.

Required?	false
Default Value	
Accept Pipeline Input?	false

-UserFilterEnabled<Boolean>

Specifies whether the session linger's user filter is enabled or disabled. Where the user filter is enabled, lingering is enabled only to users who appear in the filter (either explicitly or by virtue of group membership).

Required?	false
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.SessionLinger Session linger settings may be specified through pipeline input.

Return Values

None or Citrix.Broker.Admin.SDK.SessionLinger

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.SessionLinger object.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Set-BrokerSessionLinger -DesktopGroupName Accounting -MaxTimeBeforeDisconnect 0:10
```

Sets the disconnect time for the session linger setting associated with desktop group named Accounting.

Set-BrokerSessionMetadata

May 04, 2016

Creates/Updates metadata key-value pairs for Session

Syntax

```
Set-BrokerSessionMetadata [-SessionId] <Int64> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSessionMetadata [-SessionId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSessionMetadata [-SessionId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSessionMetadata [-InputObject] <Session[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSessionMetadata [-InputObject] <Session[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSessionMetadata [-SessionName] <Guid> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSessionMetadata [-SessionName] <Guid> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerSessionMetadata cmdlet creates/updates metadata key-value pairs for Session. The Session can be specified by InputObject or piping.

Related topics

Parameters

-SessionId<Int64>

Specifies the Session object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Session[]>

Specifies the Session objects whose Metadata is to be created/updated.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-SessionName<Guid>

Specifies the Session object whose Metadata is to be created/updated by name.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerSession You can pipe the Session to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerSession

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerSession object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerSessionMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the Session whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerSession | Set-BrokerSessionMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the Session in the site

----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerSessionMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the Session in the site whose name is 'objname'

Set-BrokerSessionPreLaunch

May 04, 2016

Updates the values of one or more desktop group session pre-launch settings.

Syntax

Set-BrokerSessionPreLaunch [-InputObject] <SessionPreLaunch[]> [-PassThru] [-Enabled <Boolean>] [-MaxAverageLoadThreshold <Int32>] [-MaxLoadPerMachineThreshold <Int32>] [-MaxTimeBeforeDisconnect <TimeSpan>] [-MaxTimeBeforeTerminate <TimeSpan>] [-UserFilterEnabled <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Set-BrokerSessionPreLaunch [-DesktopGroupName] <String> [-PassThru] [-Enabled <Boolean>] [-MaxAverageLoadThreshold <Int32>] [-MaxLoadPerMachineThreshold <Int32>] [-MaxTimeBeforeDisconnect <TimeSpan>] [-MaxTimeBeforeTerminate <TimeSpan>] [-UserFilterEnabled <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Detailed Description

The Set-BrokerSessionPreLaunch cmdlet is used to alter the settings of an existing desktop group session pre-launch setting.

Related topics

[New-BrokerSessionPreLaunch](#)

[Get-BrokerSessionPreLaunch](#)

[Remove-BrokerSessionPreLaunch](#)

Parameters

-InputObject<SessionPreLaunch[]>

The session pre-launch to be modified.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-DesktopGroupName<String>

The name of the desktop group whose session pre-launch setting is to be modified.

Required?	true

Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-Enabled<Boolean>

Boolean that indicates if the session pre-launch setting is to be enabled or disabled.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MaxAverageLoadThreshold<Int32>

Specifies the average load threshold across the desktop group. When the threshold hits, pre-launched sessions across the group be terminated to reduce load. Sessions that have been pre-launched the longest will be chosen first.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MaxLoadPerMachineThreshold<Int32>

Specifies the maximum load threshold per machine in the desktop group. When the threshold hits, pre-launched sessions on each loaded machine will be terminated to reduce load. Sessions that have been pre-launched the longest will be chosen first.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MaxTimeBeforeDisconnect<TimeSpan>

Specifies the time by which a pre-launched session will be disconnected. The disconnect time cannot be greater than the terminate timer (if enabled). When the disconnect and terminate times are the same, the terminate time takes precedence.

Required?	false
Default Value	
Accept Pipeline Input?	false

-MaxTimeBeforeTerminate<TimeSpan>

Specifies the time by which a pre-launched session will be terminated. When the disconnect and terminate times are the same, the terminate time takes precedence.

Required?	false
Default Value	
Accept Pipeline Input?	false

-UserFilterEnabled<Boolean>

Specifies whether the session pre-launch's user filter is enabled or disabled. Where the user filter is enabled, pre-launch is enabled only to users who appear in the filter (either explicitly or by virtue of group membership).

Required?	false
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop

Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.SessionPreLaunch Session pre-launch settings may be specified through pipeline input.

Return Values

None or Citrix.Broker.Admin.SDK.SessionPreLaunch

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.SessionPreLaunch object.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Set-BrokerSessionPreLaunch -DesktopGroupName Accounting -MaxTimeBeforeDisconnect 0:10
```

Sets the disconnect time for the session pre-launch setting associated with desktop group named Accounting.

Set-BrokerSharedDesktop

May 04, 2016
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

May 04, 2016

Changes the overall settings of the current XenDesktop broker site.

Syntax

```
Set-BrokerSite [-PassThru] [-BaseOU <Guid>] [-ColorDepth <ColorDepth>] [-ConnectionLeasingEnabled <Boolean>] [-DesktopGroupIconUid <Int32>] [-DnsResolutionEnabled <Boolean>] [-SecureIcaRequired <Boolean>] [-TrustRequestsSentToTheXmlServicePort <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerSite cmdlet modifies properties of the current broker site.

The broker site is a top-level, logical representation of the XenDesktop site, from the perspective of the brokering services running within the site. It defines various site-wide default attributes used by the brokering services.

A XenDesktop installation has only a single broker site instance.

Related topics

[Get-BrokerSite](#)

[New-BrokerIcon](#)

Parameters

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-BaseOU<Guid>

Changes the objectGUID property identifying the base OU in Active Directory used for desktop registrations. For sites using only registry-based discovery (the default) this value is \$null.

Any desktop attempting to register through a different OU from the one specified here is rejected. Note that desktops configured for registry-based discovery can register with the site, even if a BaseOU value is specified.

Information held in Active Directory is not modified by changing this value.

Typically, this property is changed only by using the Set-ADControllerDiscovery.ps1 script.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ColorDepth<ColorDepth>

Changes the default color depth for new desktop groups, if no color depth is specified explicitly when a group is created. Changing this default has no impact on the color depths used already by existing groups.

Valid values are FourBit, EightBit, SixteenBit, and TwentyFourBit.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ConnectionLeasingEnabled<Boolean>

Enabled or disable connection leasing on the site.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DesktopGroupIconUid<Int32>

Changes the default desktop icon used for new desktop groups if no icon is specified explicitly when a group is created. Changing this default has no impact on the icons used already by existing groups.

The specified icon must already have been added to the site using New-BrokerIcon.

Required?	false
Default Value	
Accept Pipeline Input?	false

-DnsResolutionEnabled<Boolean>

Changes whether ICA files returned by a broker service to a user device contain the numeric IP address or the DNS name of the desktop machine to which a session should be established.

With the default value (\$false), ICA files will always contain a numeric IP address. To have DNS names appear in the ICA files, set the value to \$true.

Even when DNS resolution is enabled (\$true), IP addresses may still appear in ICA files. The reasons for this include, for example, that the broker service is unable to obtain a DNS name for the target machine, or that Web Interface is configured to always use numeric IP addresses in this context.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SecureIcaRequired<Boolean>

Changes the default SecureICA usage requirements for new desktop groups if no SecureICA setting is specified explicitly when a group is created. Changing this default has no impact on the SecureICA usage requirements of existing groups.

Required?	false
Default Value	
Accept Pipeline Input?	false

-TrustRequestsSentToTheXmlServicePort<Boolean>

Changes whether the XML Service (as used by Web Interface) implicitly trusts the originator of requests it receives, or whether it fully authenticates them.

With the default value (\$false), full authentication checks are performed. However, you must enable this setting (\$true) to allow support for "Pass-through" authentication, and/or connections routed through Access Gateway.

If this setting is enabled, you must ensure that controllers running the brokering services are securely firewalled.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

None or Citrix.Broker.Admin.SDK.Site

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.Site object.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Set-BrokerSite -ColorDepth SixteenBit
```

Specifies that any new desktop groups created, where a color depth value is not specified, default to using 16-bit color depth for user sessions to desktops or applications within that group.

Set-BrokerSiteMetadata

May 04, 2016

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

May 04, 2016

Creates/Updates metadata key-value pairs for Tag

Syntax

```
Set-BrokerTagMetadata [-TagId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerTagMetadata [-TagId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerTagMetadata [-TagId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerTagMetadata [-InputObject] <Tag[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerTagMetadata [-InputObject] <Tag[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerTagMetadata [-TagName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerTagMetadata [-TagName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-BrokerTagMetadata cmdlet creates/updates metadata key-value pairs for Tag. The Tag can be specified by InputObject or piping.

Related topics

Parameters

-TagId<Int32>

Specifies the Tag object whose Metadata is to be created/updated by ID.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-InputObject<Tag[]>

Specifies the Tag objects whose Metadata is to be created/updated.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-TagName<String>

Specifies the Tag object whose Metadata is to be created/updated by name.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-Name<String>

Specifies the name of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Value<String>

Specifies the value of the Metadata member to be created/updated

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Map<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-PassThru<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Citrix.Broker.Admin.SDK.BrokerTag You can pipe the Tag to hold the new or updated metadata.

Return Values

None or Citrix.Broker.Admin.SDK.BrokerTag

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerTag object.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerTagMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the Tag whose instance is pointed by \$obj-Uid

----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerTag | Set-BrokerTagMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the Tag in the site

----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerTagMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the Tag in the site whose name is 'objname'

Start-BrokerCatalogPvdImagePrepare

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Tests whether a database is suitable for use by the Citrix Broker Service.

Syntax

```
Test-BrokerDBConnection [-DBConnection] <String> [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Tests whether the database specified in the given connection string is suitable for use by the currently selected Citrix Broker Service instance.

The service attempts to contact the specified database and returns a status indicating whether the database is both contactable and usable. The test does not impact any currently established connection from the service instance to another database in any way. The tested connection string is not recorded.

Only use of Windows authentication within the connection string is supported; a connection string containing SQL authentication credentials is always rejected as invalid.

The current service instance is the one on the local machine, or the one most recently specified using the -AdminAddress parameter of a Broker SDK cmdlet.

Related topics

[Get-BrokerServiceStatus](#)

[Get-BrokerDBConnection](#)

[Set-BrokerDBConnection](#)

Parameters

-DBConnection<String>

Specifies the database connection string to be tested by the currently selected Citrix Broker Service instance.

Required?	true
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Test-BrokerDBConnection cmdlet returns an object describing the status of the selected Broker Service instance that would result if the connection string were used with the Set-BrokerDBConnection cmdlet together with extra diagnostics information for the specified connection string. The actual current status of the service is not changed. Possible diagnostic values are:

-- OK:

The Set-BrokerDBConnection command would succeed if it were executed with the supplied connection string.

-- DBUnconfigured:

No database connection string is set for the service instance.

-- DBRejectedConnection:

The database rejected the logon attempt from the Broker Service instance. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

-- InvalidDBConfigured:

The specified database does not exist, is not visible to the Broker Service instance, or the service's schema within the database is invalid.

-- DBNotFound:

The specified database could not be located with the given connection string.

-- DBNewerVersionThanService:

The Broker Service instance is older than, and incompatible with, the service's schema in the database. The service instance needs upgrading.

-- DBOlderVersionThanService:

The Broker Service instance is newer than, and incompatible with, the service's schema in the database. The database schema needs upgrading.

-- DBVersionChangeInProgress:

A database schema upgrade is currently in progress.

-- PendingFailure:

Connectivity between the Broker Service instance and the database has been lost. This may be a transitory network error, but may indicate a loss of connectivity that requires administrator intervention.

-- Failed:

Connectivity between the Broker Service instance and the database has been lost for an extended period of time, or has failed due to a configuration problem. The service instance cannot operate while its connection to the database is unavailable.

-- Unknown:

Service status cannot be determined.

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidDBConnectionString

The database connection string has an invalid format.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

```
C:\PS>Test-BrokerDBConnection "Server=dbserver\SQLEXPRESS;Database=XDDB;Trusted_Connection=True"
```

Tests whether the service instance could use a database called XDDB on an SQL Server Express database running on the machine called dbserver. Integrated Windows authentication is required.

Test-BrokerDesktopGroupNameAvailable

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Imports or updates all of the file type associations for the specified worker.

Syntax

```
Update-BrokerImportedFTA -DesktopUids <Int32[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Imports or updates the file type associations from a specified worker machine.

File type association associates a file extension (such as ".txt") with an application (such as Notepad). In a Citrix environment file type associations on a user device can be configured so that when an user clicks on a document it launches the appropriate published application. This is known as "content redirection".

Imported file type associations are different from configured file type associations. Imported file type associations are lists of known file type associations for a given desktop group. Configured file type associations are those that are actually associated with published applications for the purposes of content redirection.

Initially the system is not aware of any extensions, and they must be imported by the Citrix administrator. To import or update file type associations from a worker machine, two criteria must be met:

- o The worker machine must not be in use
- o The worker machine must be in maintenance mode

For more information about putting a worker machine in maintenance mode, see the Set-BrokerPrivateDesktop and Set-BrokerSharedDesktop cmdlets.

Imported file type associations are grouped together based on the desktop group of the machine from which they were imported. All file types for a desktop group are deleted. There is no mechanism for deleting a subset imported file type associations for a specific desktop group.

If file type associations are imported more than once for a desktop group, for example, if this cmdlet is run twice for two workers belonging to the same desktop group, all existing imported file type associations for that desktop group are deleted and imported again.

Related topics

[Get-BrokerImportedFTA](#)

[Remove-BrokerImportedFTA](#)

Parameters

-DesktopUids<Int32[]>

Imports or updates the file type associations from the specified desktop. The desktop must belong to a desktop group of the Private or Shared desktop kind.

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

Int32[] An array of Uids for desktops can be supplied as input.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> $desktop = Get-BrokerSharedDesktop -MachineName "ACME\Worker1"
C:\PS> Set-BrokerSharedDesktop $desktop -InMaintenanceMode $true
C:\PS> Update-BrokerImportedFTA -DesktopUids $desktop.Uid
C:\PS> Set-BrokerSharedDesktop $desktop -InMaintenanceMode $false
```

Gets an object for the worker machine named "Worker1" in the "ACME" domain, and ensures no users can connect to it, before importing the file type associations from that desktop and re-enabling it.

Update-BrokerLocalLeaseCache

May 04, 2016

Flushes the local lease cache.

Syntax

Update-BrokerLocalLeaseCache [-Workers] [-Applications] [-Icons] [-Desktops] [-Leases] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Detailed Description

Removes all local cached lease data and any state information stored in the registry.

Related topics

[Remove-BrokerLease](#)

Parameters

-Workers<SwitchParameter>

Removes all locally cached workers on the controller. The worker cache will be repopulated from the current site database contents after a short delay.

Required?	false
Default Value	false
Accept Pipeline Input?	false

-Applications<SwitchParameter>

Removes all locally cached applications on the controller. The application cache will be repopulated from the current site database contents after a short delay.

Required?	false
Default Value	false
Accept Pipeline Input?	false

-Icons<SwitchParameter>

Removes all locally cached icons on the controller. The icon cache will be repopulated from the current site database contents after a short delay.

Required?	false
Default Value	false
Accept Pipeline Input?	false

-Desktops<SwitchParameter>

Removes all locally cached desktops on the controller. The desktop cache will be repopulated from the current site database contents after a short delay.

Required?	false
Default Value	false
Accept Pipeline Input?	false

-Leases<SwitchParameter>

Removes all locally cached leases on the controller. The lease cache will be repopulated from the current site database contents after a short delay.

Required?	false
Default Value	false
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

Input Type

None

Return Values

None

Notes

The local cache for lease and other data like worker, desktop, application and icon information is removed and will be repopulated from the current site database contents after a short delay.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Update-BrokerLocalLeaseCache
```

Flushes the local lease cache for all objects and deletes any state information stored in the registry.

----- **EXAMPLE 2** -----

```
C:\PS> Update-BrokerLocalLeaseCache -Workers
```

Flushes the local lease cache for all workers and deletes any state information stored in the registry.

Update-BrokerNameCache

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016
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

May 04, 2016
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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Lists the site's supported product names and codes.

Syntax

```
Get-ConfigProduct [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Related topics

[Get-ConfigSite](#)

[Set-ConfigSite](#)

Parameters

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

PObject

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-ConfigProduct
```

Lists the supported products by name and code.

Get-ConfigProductEdition

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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_Config_Filtering for details.

Required?	false
Default Value	false
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

See about_Config_Filtering for details.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

See about_Config_Filtering for details.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

See about_Config_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Filter<String>

See about_Config_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	LocalHost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

Citrix.Configuration.Sdk.ServiceInstance

This represents a service instance and has the following parameters;

ServiceGroupUid <Guid>

The unique identifier for the service group to which the service instance belongs.

ServiceGroupName <string>

The name of the service group to which the service instance belongs.

ServiceInstanceUid <Guid>

The unique identifier for the service instance.

ServiceType <string>

The type of the service group.

Address <string>

The contact address for the service instance.

Binding <string>

The binding to use for connections to the service instance.

Version <int>

The version of the service instance.

ServiceAccount <string>

The AD computer account for the computer that is providing the service instance.

ServiceAccountSid <string>

The AD computer account SID for the computer that is providing the service instance.

InterfaceType <string>

The interface type for the service instance.

Metadata <Citrix.Configuration.Sdk.Metadata[]>

The metadata for the service instance.

Notes

In the case of failure, the following errors can result.

Error Codes ----- PartialData Only a subset of the available data was returned. CouldNotQueryDatabase The query required to get the database was not defined. CommunicationError An error occurred while communicating with the service. DatabaseNotConfigured The operation could not be completed because the database for the service is not configured. InvalidFilter A filtering expression was supplied that could not be interpreted for this cmdlet. ExceptionThrown An unexpected error occurred. To locate more details see the Windows event logs on the controller being used, or examine the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

C:\>Get-ConfigRegisteredServiceInstance

Address : http://MyServer.com:80/Citrix/MyContract/v1
Binding : wcf_HTTP_kerb

InterfaceType : SDK
Metadata : {}
MetadataMap : {}
ServiceAccount : ENG\MyAccount\$
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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Gets the service groups that match the parameters supplied.

Syntax

```
Get-ConfigServiceGroup [-ServiceGroupUid <Guid>] [-ServiceGroupName <String>] [-ServiceType <String>] [-Metadata <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Use this cmdlet to retrieve existing service groups that match the parameters supplied. If no parameters are supplied, all the service groups are returned.

Related topics

Parameters

-ServiceGroupUid<Guid>

The unique identifier for the service group.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ServiceGroupName<String>

Required?	false
Default Value	
Accept Pipeline Input?	false

-ServiceType<String>

The service type for the service group.

Required?	false
Default Value	

Accept Pipeline Input?	false

-Metadata<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

See about_Config_Filtering for details.

Required?	false
Default Value	false
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

See about_Config_Filtering for details.

Required?	false

Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

See about_Config_Filtering for details.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

See about_Config_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Filter<String>

See about_Config_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false

Default Value	LocalHost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

Citrix.Configuration.Sdk.ServiceGroup

This represents a service instance and has the following parameters;

ServiceGroupUid <Guid>

The unique identifier for the service group.

ServiceGroupName <string>

The name of the service group.

ServiceType <string>

The type of the service group.

Metadata <Citrix.Configuration.Sdk.Metadata[]>

The metadata for the service group.

Notes

In the case of failure, the following errors can result.

Error Codes -----

DatabaseNotConfigured The operation could not be completed because the database for the service is not configured.
 PartialData Only a subset of the available data was returned. CouldNotQueryDatabase The Query required to get the database was not defined. CommunicationError An error occurred while communicating with the service. InvalidFilter A filtering expression was supplied that could not be interpreted for this cmdlet. ExceptionThrown An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

C:\>Get-ConfigServiceGroup

Return all the service groups that are registered in the Configuration Service.

----- EXAMPLE 2 -----

C:\>Get-ConfigRegisteredServiceInstance -ServiceType "config"

Return all the service groups that are registered in the Configuration Service and are of type 'config' (i.e. the service groups for the Configuration Service).

Get-ConfigServiceInstance

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Allows the registration of a service instance.

Syntax

```
Register-ConfigServiceInstance -ServiceGroupUid <Guid> -ServiceGroupName <String> -ServiceType <String> -Address <String> -Binding <String> -Version <Int32> -ServiceAccountSid <String> -InterfaceType <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Register-ConfigServiceInstance -ServiceGroupUid <Guid> -ServiceGroupName <String> -ServiceType <String> -Address <String> -Binding <String> -Version <Int32> -ServiceAccount <String> -InterfaceType <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Register-ConfigServiceInstance -ServiceInstance <ServiceInstance[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Use this cmdlet to register service instance items in the Configuration Service. Service instances can be registered either by retrieving the data directly from other services or by manually entering the details into this command.

If the service group specified by the service instance already exists, the service is added to the service group, otherwise a new service group is created to hold the service instance.

Related topics

[Unregister-ConfigRegisteredServiceInstance](#)

[Add-ConfigRegisteredServiceInstanceMetadata](#)

[Set-ConfigRegisteredServiceInstanceMetadata](#)

[Remove-ConfigRegisteredServiceInstanceMetadata](#)

[Add-ConfigServiceGroupMetadata](#)

[Set-ConfigServiceGroupMetadata](#)

[Remove-ConfigServiceGroupMetadata](#)

Parameters

-ServiceGroupUid<Guid>

The Service Group Unique Identifier

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-ServiceGroupName<String>

The name of the service group

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-ServiceType<String>

The type of the service group

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Address<String>

The address that is used to access the service instance.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Binding<String>

The binding type that must be used to access the service instance.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Version<Int32>

The version of the service instance.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-ServiceAccountSid<String>

The AD computer account Sid for the computer on which the service instance is hosted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-InterfaceType<String>

The type of interface that the service provides (i.e. SDK, InterService, Peer).

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-ServiceAccount<String>

The AD computer account name (domain qualified) for the computer on which the service instance is hosted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-ServiceInstance<ServiceInstance[]>

The service instances to register.

Required?	true

Default Value	
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the logging id of the high-level operation that this cmdlet is part of.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-AdminAddress<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	LocalHost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Input Type

Citrix.Configuration.Sdk.ServiceInstance An object with the following parameters can be used to register a service instance. Address, ServiceGroupUid, ServiceGroupName, ServiceType, Binding, Version, InterfaceType.

Return Values

Citrix.Configuration.Sdk.ServiceInstance

This represents a service instance and has the following parameters;

ServiceGroupUid <Guid>

The unique identifier for the service group to which the service instance belongs.

ServiceGroupName <string>

The name of the service group to which the service instance belongs.

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016
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

May 04, 2016
Adds or updates metadata on the given ServiceInstance.

Syntax

```
Set-ConfigRegisteredServiceInstanceMetadata [-ServiceInstanceUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Set-ConfigRegisteredServiceInstanceMetadata [-ServiceInstanceUid] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]

Set-ConfigRegisteredServiceInstanceMetadata [-InputObject] <ServiceInstance[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]

Set-ConfigRegisteredServiceInstanceMetadata [-InputObject] <ServiceInstance[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]
```

Detailed Description

Provides the ability for additional custom data to be stored against given ServiceInstance objects.

Related topics

- [Add-ConfigRegisteredServiceInstanceMetadata](#)
- [Remove-ConfigRegisteredServiceInstanceMetadata](#)

Parameters

-ServiceInstanceUid<Guid>

Id of the ServiceInstance

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<ServiceInstance[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true

Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the property name of the metadata to be added. The property must be unique for the ServiceInstance specified. The property cannot contain any of the following characters \,;,#.*?=<>|[]()''

Required?	true
Default Value	
Accept Pipeline Input?	false

-Value<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of 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

May 04, 2016

Adds or updates metadata on the given ServiceGroup.

Syntax

```
Set-ConfigServiceGroupMetadata [-ServiceGroupUid] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ConfigServiceGroupMetadata [-ServiceGroupUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ConfigServiceGroupMetadata [-InputObject] <ServiceGroup[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ConfigServiceGroupMetadata [-InputObject] <ServiceGroup[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Provides the ability for additional custom data to be stored against given ServiceGroup objects.

Related topics

[Add-ConfigServiceGroupMetadata](#)

[Remove-ConfigServiceGroupMetadata](#)

Parameters

-ServiceGroupUid<Guid>

Id of the ServiceGroup

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<ServiceGroup[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the property name of the metadata to be added. The property must be unique for the ServiceGroup specified. The property cannot contain any of the following characters \/:;#.*?=<>|[]()"

Required?	true
Default Value	
Accept Pipeline Input?	false

-Value<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

System.Collections.Generic.Dictionary[String,String]

Set-ConfigServiceGroupMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-ConfigServiceGroupMetadata -ServiceGroupUid 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

Key	Value
---	-----
property	value

Add metadata with a name of 'property' and a value of 'value' to the ServiceGroup with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

Set-ConfigServiceMetadata

May 04, 2016

Adds or updates metadata on the given Service.

Syntax

```
Set-ConfigServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]
```

```
Set-ConfigServiceMetadata [-ServiceHostId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]
```

```
Set-ConfigServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]
```

```
Set-ConfigServiceMetadata [-InputObject] <Service[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]
```

Detailed Description

Allows you to store additional custom data against given Service objects.

Related topics

[Remove-ConfigServiceMetadata](#)

Parameters

-ServiceHostId<Guid>

Id of the Service

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Service[]>

Objects to which metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Map<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-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

May 04, 2016

Changes the overall settings of the site.

Syntax

```
Set-ConfigSite [-SiteName <String>] [-ProductCode <String>] [-ProductEdition <String>] [-ProductVersion <String>] [-LicensingModel <String>] [-LicenseServerName <String>] [-LicenseServerPort <Int32>] [-LicenseServerUri <Uri>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Set-ConfigSite cmdlet modifies properties of the site.

The site is a top-level, logical representation of the XenDesktop site, from the perspective of the configuration services running within the site.

A XenDesktop installation has only a single site instance.

Modifications to the product code, product edition, product version and licensing model properties are successful only if their values are consistent with the feature table. Use the Get-ConfigProduct, Get-ConfigProductEdition, Get-ConfigProductVersion and Get-ConfigLicensingModel cmdlets to determine consistent values.

To configure the site, first import the feature table using the Import-ConfigFeatureTable cmdlet.

Related topics

[Export-ConfigFeatureTable](#)

[Get-ConfigSite](#)

[Get-ConfigProduct](#)

[Get-ConfigProductEdition](#)

[Get-ConfigProductFeature](#)

[Get-ConfigProductVersion](#)

[Get-ConfigLicensingModel](#)

Parameters

-SiteName<String>

Changes the name of the site.

Required?	false
Default Value	

Accept Pipeline Input?	false

-ProductCode<String>

Changes the product code.

The Get-ConfigProduct cmdlet returns a list of supported values.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ProductEdition<String>

Changes the license edition. A license matching the specified edition must be available within the site's license server.

The Get-ConfigProductEdition returns a list of supported values.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ProductVersion<String>

Changes the product version.

The Get-ConfigProductVersion returns a list of supported values.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LicensingModel<String>

Changes the license model. A license matching the specified model must be available within the site's license server.

The Get-ConfigLicensingModel returns a list of supported values.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LicenseServerName<String>

Changes the machine used by the brokering services to obtain licenses for desktop and application session brokering. The specified machine must be running a Citrix license server and have suitable licenses installed.

The license server machine can be specified by its DNS name ('machine.domain') or its numeric IP address.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LicenseServerPort<Int32>

Changes the port number on the license server machine used by the brokering services to contact the Citrix license server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LicenseServerUri<Uri>

Changes the Uri of the web server for licensing. The hostname component of this Uri must match the Site's LicenseServerName property.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

Site

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Set-ConfigSite -ProductEdition PLT
```

Specifies the use of a Platinum edition license. A suitable license must be available on the site's license server.

Set-ConfigSiteMetadata

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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.Timespan>

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Gets high level operations

Syntax

```
Get-LogHighLevelOperation [-Id <Guid>] [-Text <String>] [-StartTime <DateTime>] [-Source <String>] [-EndTime <DateTime>]  
[-IsSuccessful <Boolean>] [-User <String>] [-AdminMachineIP <String>] [-OperationType <OperationType>] [-TargetType  
<String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-  
Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Retrieves high level operations matching the specified criteria. If no parameters are specified this cmdlet returns all high level operations.

Related topics

[Start-LogHighLevelOperation](#)

[Stop-LogHighLevelOperation](#)

[Get-LogLowLevelOperation](#)

Parameters

-Id<Guid>

Gets the high level operation with the specified identifier.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Text<String>

Gets high level operations with the specified text

Required?	false
Default Value	
Accept Pipeline Input?	false

-StartTime<DateTime>

Gets high level operations with the specified start time

Required?	false
Default Value	
Accept Pipeline Input?	false

-Source<String>

Gets high level operations with the specified source.

Required?	false
Default Value	
Accept Pipeline Input?	false

-EndTime<DateTime>

Gets high level operations with the specified end time.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IsSuccessful<Boolean>

Gets high level operations with the specified success indicator.

Required?	false
Default Value	
Accept Pipeline Input?	false

-User<String>

Gets high level operations logged by the specified user.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminMachineIP<String>

Gets high level operations logged from the machine with the specified IP address.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OperationType<OperationType>

Gets high level operations with the specified operation type. Values can be:

- o AdminActivity - to get operations which log administration activity.
- o ConfigurationChange - to get operations which log configuration changes.

Required?	false
Default Value	
Accept Pipeline Input?	false

-TargetType<String>

Gets high level operations with the specified target type. The target type describes the type of object that was the target of the configuration change. For example, "Session" or "Machine".

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about_Log_Filtering for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	

Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell-style filter expression. See about_Log_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

Citrix.ConfigurationLogging.Sdk.HighLevelOperation

The returned HighLevelOperation object has the following properties:

- o Id - The unique identifier of the operation.
- o Text - A description of the operation.

- o StartTime - The date and time that the operation started.
- o EndTime - The date and time that the operation completed. This will be null if the operation is still in progress, or if the operation never completed.
- o IsSuccessful - Indicates whether the operation completed successfully or not. This will be null if the operation is still in progress, or if the operation never completed.
- o User - The identifier of the administrator who performed the operation.
- o AdminMachineIP - The IP address of the machine that the operation was initiated from.
- o Source - Identifies the XenDesktop console, or custom script, where the operation was initiated from. For example, "Desktop Studio", "Desktop Director", "My custom script".
- o OperationType - The operation type. This can be 'AdminActivity' or 'ConfigurationChange'.
- o TargetTypes - Identifies the type of objects that were affected by the operation. For example, "Catalog" or "Desktop","Machine".
- o Parameters - The names and values of the parameters that were supplied for the operation.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Get-LogHighLevelOperation
Get all high level operations
```

----- EXAMPLE 2 -----

```
C:\PS> Get-LogHighLevelOperation -OperationType ConfigurationChange
Get high level operations which log configuration changes.
```

----- EXAMPLE 3 -----

```
C:\PS> Get-LogHighLevelOperation -OperationType AdminActivity
Get high level operations which log administration activities.
```

----- EXAMPLE 4 -----

```
C:\PS> Get-LogHighLevelOperation -Filter{ StartTime -ge "2013-02-27 09:00:00" -and EndTime -le "2013-02-27 18:00:00"}
Use advanced filtering to get high level operations with a start time on or after "2013-02-27 09:00:00" and an end time on or before "2013-02-27 18:00:00".
```

----- EXAMPLE 5 -----

```
C:\PS> Get-LogHighLevelOperation -EndTime $null
C:\PS> Get-LogHighLevelOperation -IsSuccessful $null
Either of these commands will get high level operations which have not yet been completed.
```

----- EXAMPLE 6 -----

```
C:\PS> Get-LogHighLevelOperation -User "DOMAIN\UserName"
```

Get high level operations performed by user "DOMAIN\UserName".

Get-LogInstalledDBVersion

May 04, 2016

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

May 04, 2016

Gets low level operations

Syntax

```
Get-LogLowLevelOperation [-Id <Guid>] [-HighLevelOperationId <Guid>] [-StartTime <DateTime>] [-EndTime <DateTime>] [-IsSuccessful <Boolean>] [-User <String>] [-AdminMachineIP <String>] [-Text <String>] [-Source <String>] [-SourceSdk <String>] [-OperationType <OperationType>] [-TargetType <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Retrieves low level operations matching the specified criteria. If no parameters are specified this cmdlet returns all low level operations.

Related topics

[Get-LogLowLevelOperation](#)

Parameters

-Id<Guid>

Gets the low level operation with the specified identifier.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-HighLevelOperationId<Guid>

Gets low level operations for the high level operation with the specified identifier.

Required?	false
Default Value	
Accept Pipeline Input?	false

-StartTime<DateTime>

Gets low level operations with the specified start time

Required?	false
Default Value	
Accept Pipeline Input?	false

-EndTime<DateTime>

Gets low level operations with the specified end time.

Required?	false
Default Value	
Accept Pipeline Input?	false

-IsSuccessful<Boolean>

Gets low level operations with the specified success indicator.

Required?	false
Default Value	
Accept Pipeline Input?	false

-User<String>

Gets low level operations logged by the specified administrator.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminMachineIP<String>

Gets low level operations logged from the machine with the specified IP address.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Text<String>

Gets low level operations with the specified text

Required?	false
Default Value	
Accept Pipeline Input?	false

-Source<String>

Gets low level operations with the specified source.

Required?	false
Default Value	
Accept Pipeline Input?	false

-SourceSdk<String>

Gets low level operations logged from the SDK with the specified identifier.

Required?	false
Default Value	
Accept Pipeline Input?	false

-OperationType<OperationType>

Gets low level operations with the specified operation type. Values can be:

o AdminActivity - to get operations which log administration activity.

o ConfigurationChange - to get operations which log configuration changes.

Required?	false
Default Value	
Accept Pipeline Input?	false

-TargetType<String>

Gets low level operations with the specified target type. The target type describes the type of object that was the target of the configuration change. For example, "Session" or "Machine".

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about_Log_Filtering for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell-style filter expression. See about_Log_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

Citrix.ConfigurationLogging.Sdk.LowLevelOperation

The returned LowLevelOperation object has the following properties:

- o Id - The unique identifier of the operation.
- o Text - A description of the operation.
- o HighLevelOperationId - The unique identifier of the related high level operation.
- o StartTime - The date and time that the operation started.
- o EndTime - The date and time that the operation completed. This will be null if the operation is still in progress, or if the operation never completed.
- o IsSuccessful - Indicates whether the operation completed successfully or not. This will be null if the operation is still in progress, or if the operation never completed.
- o AdminSid - The identifier of the administrator who performed the operation.
- o AdminMachineIP - The IP address of the machine that the operation was performed on.
- o Source - The name of the XenDesktop service that the operation was performed on; for example, "MachineCreation", "DelegatedAdmin".
- o SourceSdk - The identifier of the XenDesktop service SDK through which the operation was performed; for example, "Prov", "Admin".
- o OperationType - The operation type. This can be 'AdminActivity' or 'ConfigurationChange'.
- o TargetTypes - Identifies the type of objects that were affected by the operation. For example, "Catalog" or "Desktop","Machine".
- o Parameters - The names and values of the parameters that were supplied for the operation.
- o Details - A collection of OperationDetail objects containing specific information about each object affected by the operation.

Each OperationDetail object in the 'Details' collection has the following properties:

- o TargetUid - The unique identifier of the target object affected by the operation.
- o TargetName - The name of the target object affected by the operation.
- o TargetType - The type of the target object.
- o Text - The description of operation performed on the target object.
- o StartTime - The date and time that the operation started.
- o EndTime - The date and time that the operation completed. This will be null if the operation is still in progress, or if the operation never completed.
- o IsSuccessful - Indicates whether the operation completed successfully or not. This will be null if the operation is still in progress, or if the operation didn't complete.

The following properties will be set if the operation changed a property on the object:

- o PropertyName - The name of the changed property.
- o NewValue - The new property value.
- o PreviousValue - The previous property value.
- o AddValue - If the object property contains a set of values, this specifies the new value which was added to the set.
- o RemoveValue - If the object property contains a set of values, this specifies the value which was removed from the set.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Get-LogLowLevelOperation
Get all low level operations
```

----- EXAMPLE 2 -----

```
C:\PS> Get-LogLowLevelOperation -OperationType ConfigurationChange
Get low level operations which log configuration changes.
```

----- EXAMPLE 3 -----

```
C:\PS> Get-LogLowLevelOperation -OperationType AdminActivity
Get low level operations which log administration activities.
```

----- EXAMPLE 4 -----

```
C:\PS> Get-LogLowLevelOperation -Filter{ StartTime -ge "2013-02-27 09:00:00" -and EndTime -le "2013-02-27 18:00:00"}
Use advanced filtering to get low level operations with a start time on or after "2013-02-27 09:00:00" and an end time on or before "2013-02-27 18:00:00".
```

----- EXAMPLE 5 -----

```
C:\PS> Get-LogLowLevelOperation -EndTime $null
```

```
C:\PS> Get-LogLowLevelOperation -IsSuccessful $null
```

Either of these commands will get low level operations which have not yet been completed.

----- **EXAMPLE 6** -----

```
C:\PS> Get-LogLowLevelOperation -User "DOMAIN\UserName"
```

Get low level operations performed by user "DOMAIN\UserName".

Get-LogService

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Deletes configuration logs

Syntax

Remove-LogOperation -DatabaseCredentials <PSCredential> [-StartDateRange <DateTime>] [-EndDateRange <DateTime>] [-IncludeIncomplete] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Remove-LogOperation -UserName <String> [-Password <String>] [-StartDateRange <DateTime>] [-EndDateRange <DateTime>] [-IncludeIncomplete] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Remove-LogOperation -UserName <String> -SecurePassword <SecureString> [-StartDateRange <DateTime>] [-EndDateRange <DateTime>] [-IncludeIncomplete] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Detailed Description

Remove-LogOperation deletes logs from the Configuration Logging database. This cmdlet targets high level operation logs for deletion. The associated low level operations logs are deleted as part of this operation via cascade deletion functionality present in the configuration logging database schema.

Related topics

Parameters

-DatabaseCredentials<PSCredential>

Specifies the credentials of a database user with permission to delete records from the Configuration Logging database.

Required?	true
Default Value	
Accept Pipeline Input?	false

-UserName<String>

Specifies the user name of a database user with permission to delete records from the Configuration Logging database.

Required?	true
Default Value	
Accept Pipeline Input?	false

-SecurePassword<SecureString>

Specifies the password a database user, in secure string form, with permission to delete records from the Configuration Logging database.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Password<String>

Specifies the password of a database user with permission to delete records from the Configuration Logging database.

Required?	false

Default Value	
Accept Pipeline Input?	false

-StartDateRange<DateTime>

Specifies the start time of the earliest high level operation to delete

Required?	false
Default Value	DateTime.Min
Accept Pipeline Input?	false

-EndDateRange<DateTime>

Specifies the end time of the latest high level operation to delete

Required?	false
Default Value	DateTime.UtcNow
Accept Pipeline Input?	false

-IncludeIncomplete<SwitchParameter>

Specifies if incomplete high level operations should be deleted.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-LogOperation -UserName "DOMAIN\User" -Password "UserPassword"
Remove all completed operation logs
```

----- EXAMPLE 2 -----

```
C:\PS> Remove-LogOperation -UserName "DOMAIN\User" -Password "UserPassword" -IncludeIncomplete
Remove all operations
```

----- EXAMPLE 3 -----

```
C:\PS> Remove-LogOperation -username "domain\userName" -password "password" -StartDateRange "2013-01-01 12:00:00" -EndDateRange "2013-01-31 12:00:00"
Delete logs started on or after "2013-01-01 12:00:00" and completed on or before "2013-01-31 12:00:00"
```

----- EXAMPLE 4 -----

```
C:\PS> $securePassword = ConvertTo-SecureString -String "UserPassword" -AsPlainText -Force
C:\PS> $credentials = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList "DOMAIN\UserName", $securePassword
C:\PS> Remove-LogOperation -StartDateRange -DatabaseCredentials $credentials
Delete logs by supplying database user credentials via a credentials object.
```


Remove-LogServiceMetadata

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Adds or updates metadata on the given Service.

Syntax

```
Set-LogServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-LogServiceMetadata [-ServiceHostId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Set-LogServiceMetadata [-InputObject] <Service[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Set-LogServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

Allows you to store additional custom data against given Service objects.

Related topics

[Remove-LogServiceMetadata](#)

Parameters

-ServiceHostId<Guid>

Id of the Service

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Service[]>

Objects to which metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the property name of the metadata to be added. The property must be unique for the Service specified. The property cannot contain any of the following characters \/:;#.*?=<>|[]{}"

Required?	true
Default Value	
Accept Pipeline Input?	false

-Value<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

System.Collections.Generic.Dictionary[String,String]

Set-LogServiceMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

```
c:\PS>Set-LogServiceMetadata -ServiceHostId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

Key	Value
---	-----
property	value

Add metadata with a name of 'property' and a value of 'value' to the Service with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

Set-LogSite

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016
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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

TOPIC

about_AdminDelegatedAdminSnapin

SHORT DESCRIPTION

The Delegated Administration Service PowerShell snap-in provides administrative functions for the Delegated Administration Service.

COMMAND PREFIX

All commands in this snap-in are prefixed with 'Admin'.

LONG DESCRIPTION

The Delegated Administration Service PowerShell snap-in enables both local and remote administration of the Delegated Administration Service.

The Delegated Administration Service (or DAS for short) stores information about Citrix administrators and the rights they have. Services in the XenDesktop deployment use the DAS to determine whether a particular user has the privilege to perform an operation or not.

The snap-in provides storage and configuration of these entities:

Administrators

Each administrator object represents an individual person or a group of people identified by their Active Directory account. Administrators can be enabled and disabled.

The effective rights that a user has is the union of any rights that they have by looking at their Active Directory group membership. Disabled administrator entries are ignored for this calculation.

Once a site is setup, there must always be a full administrator and the Delegated Administration snap-in rejects requests to remove or disable the last full administrator.

Roles

A role represents a job function. That is, anyone with a given role is expected to be able to use or perform the tasks, wizards, and actions associated with that role. Administrators may have multiple roles for a particular site.

Some roles are built-in, and some editions of the product allow custom roles to be created with different combinations of permissions.

Scopes

Scopes represent a collection of objects, and are used to group objects for administrative purposes in a way that is relevant to the organisation. They can be used to represent both hierarchical and non-hierarchical relationships.

Objects can exist in multiple scopes at once. You may find it easier to think of scopes as labels, or a non-exclusive grouping such as a play-list.

All objects are implicitly in the built-in 'All' scope.

Some objects are not scoped, and access to them is through either the 'All' scope or indirectly through a scoped object. For example sessions are not directly scoped but can be accessed using the scope of the desktop group.

The DAS stores information about scopes, but the mapping between scopes and objects is stored and updated using the PowerShell snap-ins of each corresponding service. For example, Delivery Group scopes are managed using the Broker PowerShell snap-in.

Rights

Rights determine what an administrator can do and where they can do it. They are expressed as a number of <role, scope> pairs associated with each administrator.

To gain access to any particular object, a person must match an administrator object that has an appropriate right that allows the required operation in a scope that the object is a member of.

Permissions

Each task, wizard or action in the Citrix Studio or Director consoles represents a unit of functionality that an administrator can perform. Permissions are expressed at a high level and generally correspond directly to the labels in the consoles. For example: "Edit catalog", or "Create delivery group".

Permission groups:

Permissions are grouped into related functionality when displayed by the console.

Operations

Operations are the indivisible unit of functionality that each XenDesktop service can perform, and usually correspond to individual cmdlets. Internally, each permission requires a number of operations to be performed, possibly by different services.

about_Admin_Filtering

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016
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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Gets the set of Right objects associated with the current user.

Syntax

```
Get-AdminEffectiveRight [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Get-AdminEffectiveRight cmdlet returns the effective rights of the current user. This is the union of all rights of the enabled administrators that the current user matches, taking into account Active Directory group membership.

Related topics

[Get-AdminAdministrator](#)

[Add-AdminRight](#)

[Remove-AdminRight](#)

Parameters

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.DelegatedAdmin.Sdk.Right

The Rights associated with the current user

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-AdminEffectiveRight
```

Return the effective rights for the current user.

Get-AdminInstalledDBVersion

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016

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

May 04, 2016

Gets scopes configured for this site.

Syntax

```
Get-AdminScope [[-Name] <String>] [-Id <Guid>] [-BuiltIn <Boolean>] [-Description <String>] [-Metadata <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Retrieves scopes matching the specified criteria. If no parameters are specified this cmdlet enumerates all scopes.

There is one special built-in scope, the 'All' scope.

To determine what objects are currently in a scope, use the Get-<Prefix>ScopedObject from each of the relevant PowerShell snap-ins.

See about_Admin_Filtering for information about advanced filtering options.

Related topics

[New-AdminScope](#)

[Set-AdminScope](#)

[Rename-AdminScope](#)

[Remove-AdminScope](#)

[Set-AdminScopeMetadata](#)

[Remove-AdminScopeMetadata](#)

Parameters

-Name<String>

Gets scopes with the specified name.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-Id<Guid>

Gets the scope with the specified identifier.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-BuiltIn<Boolean>

Gets scopes with the specified value of the BuiltIn flag.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Description<String>

Gets scopes with the specified description.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Metadata<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about_Admin_Filtering for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0

Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell-style filter expression. See [about_Admin_Filtering](#) for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.DelegatedAdmin.Sdk.Scope

Get-AdminScope returns an object for each matching scope.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Get-AdminScope -Name *Sales*
```

List all scopes that contain the word 'Sales'.

----- EXAMPLE 2 -----

```
C:\PS> Get-AdminScope -Id 21862daf-e529-4553-ba6e-f7543217111e
```

Gets the details of the scope with the specific id.

Get-AdminService

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
Removes metadata from the given Administrator.

Syntax

Remove-AdminAdministratorMetadata -AdministratorSid <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Remove-AdminAdministratorMetadata -AdministratorSid <String> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Remove-AdminAdministratorMetadata [-AdministratorName] <String> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Remove-AdminAdministratorMetadata [-AdministratorName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Remove-AdminAdministratorMetadata [-InputObject] <Administrator[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Remove-AdminAdministratorMetadata [-InputObject] <Administrator[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Detailed Description

Provides the ability to remove metadata from the given Administrator.

Related topics

[Set-AdminAdministratorMetadata](#)

Parameters

-AdministratorName<String>

Name of the Administrator

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Administrator[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-AdministratorSid<String>

Sid of the Administrator

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-

LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-AdminAdministrator | % { Remove-AdminAdministratorMetadata -Map $_.MetadataMap }  
Remove all metadata from all Administrator objects.
```

Remove-AdminPermission

May 04, 2016

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

May 04, 2016

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

May 04, 2016
Removes a role from the site.

Syntax

```
Remove-AdminRole [-InputObject] <Role[]> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]  
Remove-AdminRole [-Id] <Guid[]> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]  
Remove-AdminRole [-Name] <String[]> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-AdminRole cmdlet deletes roles from the site.

You cannot remove built-in roles.

An error will be produced if the role being removed is currently assigned to an administrator unless you specify the -Force option. When -Force is specified, any rights that reference the role are also removed.

Related topics

- [New-AdminRole](#)
- [Get-AdminRole](#)
- [Set-AdminRole](#)
- [Rename-AdminRole](#)
- [Set-AdminRoleMetadata](#)
- [Remove-AdminRoleMetadata](#)

Parameters

-InputObject<Role[]>

Specifies the roles to remove (by role object).

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Id<Guid[]>

Specifies the roles to remove (by role id).

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Name<String[]>

Specifies the roles to remove (by role name).

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Force<SwitchParameter>

Allow removal of roles that are still in use.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name

or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Input Type

Citrix.DelegatedAdmin.Sdk.Role You can pipe the roles to be deleted into this command.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-AdminRole -Name Supervisor  
Remove the Supervisor role.
```

----- EXAMPLE 2 -----

```
C:\PS> Get-AdminRole -BuiltIn $false | Remove-AdminRole  
Attempt to remove all custom roles. This fails if one of the roles is assigned to an administrator.
```

Remove-AdminRoleMetadata

May 04, 2016

Removes metadata from the given Role.

Syntax

```
Remove-AdminRoleMetadata [-RoleId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminRoleMetadata [-RoleId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminRoleMetadata [-RoleName] <String> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminRoleMetadata [-RoleName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminRoleMetadata [-InputObject] <Role[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminRoleMetadata [-InputObject] <Role[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

Provides the ability to remove metadata from the given Role.

Related topics

[Set-AdminRoleMetadata](#)

Parameters

-RoleId<Guid>

Id of the Role

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-RoleName<String>

Name of the Role

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Role[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-

LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-AdminRole | % { Remove-AdminRoleMetadata -Map $_.MetadataMap }  
Remove all metadata from all Role objects.
```

Remove-AdminScope

May 04, 2016

Removes a scope from the site.

Syntax

```
Remove-AdminScope [-InputObject] <Scope[]> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminScope [-Id] <Guid[]> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AdminScope [-Name] <String[]> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-AdminScope cmdlet deletes scopes from the site.

You cannot remove the built-in 'All' scope.

An error will be produced if the scope being removed is currently assigned to an administrator unless you specify the -Force option. When -Force is specified, any rights that reference the scope are also removed.

Related topics

[New-AdminScope](#)

[Get-AdminScope](#)

[Set-AdminScope](#)

[Rename-AdminScope](#)

[Set-AdminScopeMetadata](#)

[Remove-AdminScopeMetadata](#)

Parameters

-InputObject <Scope[]>

Specifies the scopes to remove (by scope object).

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Id <Guid[]>

Specifies the scopes to remove (by scope id).

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Name<String[]>

Specifies the scopes to remove (by scope name).

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Force<SwitchParameter>

Allow removal of scopes that are still in use.

Required?	false
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Input Type

Citrix.DelegatedAdmin.Sdk.Scope You can pipe the scopes to be deleted into this command.

Return Values

None

Examples

----- EXAMPLE 1 -----

```
C:\PS> Remove-AdminScope -Name Sales
```

Remove the Sales scope.

----- EXAMPLE 2 -----

```
C:\PS> Get-AdminScope -BuiltIn $false | Remove-AdminScope
```

Attempt to remove all scopes (excluding the built-in 'All' scope). This fails if one of the scopes is assigned to an administrator.

Remove-AdminScopeMetadata

May 04, 2016

Removes metadata from the given Scope.

Syntax

```
Remove-AdminScopeMetadata [-ScopeId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminScopeMetadata [-ScopeId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminScopeMetadata [-ScopeName] <String> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminScopeMetadata [-ScopeName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminScopeMetadata [-InputObject] <Scope[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminScopeMetadata [-InputObject] <Scope[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

Provides the ability to remove metadata from the given Scope.

Related topics

[Set-AdminScopeMetadata](#)

Parameters

-ScopeId<Guid>

Id of the Scope

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-ScopeName<String>

Name of the Scope

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Scope[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-

LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-AdminScope | % { Remove-AdminScopeMetadata -Map $_.MetadataMap }  
Remove all metadata from all Scope objects.
```

Remove-AdminServiceMetadata

May 04, 2016

Removes metadata from the given Service.

Syntax

```
Remove-AdminServiceMetadata [-ServiceHostId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminServiceMetadata [-InputObject] <Service[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

Provides the ability to remove metadata from the given Service.

Related topics

[Set-AdminServiceMetadata](#)

Parameters

-ServiceHostId<Guid>

Id of the Service

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Service[]>

Objects to which metadata is to be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false

Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-AdminService | % { Remove-AdminServiceMetadata -Map $_.MetadataMap }
```

Remove all metadata from all Service objects.

Rename-AdminRole

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
Adds or updates metadata on the given Administrator.

Syntax

Set-AdminAdministratorMetadata -AdministratorSid <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Set-AdminAdministratorMetadata -AdministratorSid <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Set-AdminAdministratorMetadata [-AdministratorName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Set-AdminAdministratorMetadata [-AdministratorName] <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Set-AdminAdministratorMetadata [-InputObject] <Administrator[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Set-AdminAdministratorMetadata [-InputObject] <Administrator[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Detailed Description

Provides the ability for additional custom data to be stored against given Administrator objects.

Related topics

[Remove-AdminAdministratorMetadata](#)

Parameters

-AdministratorName<String>

Name of the Administrator

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Administrator[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-AdministratorSid<String>

Sid of the Administrator

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Name<String>

Specifies the property name of the metadata to be added. The property must be unique for the Administrator specified. The property cannot contain any of the following characters \/:#.*?=<>|[]{}"

Required?	true
Default Value	
Accept Pipeline Input?	false

-Value<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

System.Collections.Generic.Dictionary[String,String]

Set-AdminAdministratorMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

```
c:\PS>Set-AdminAdministratorMetadata -AdministratorSid S-1-5-21-1505241163-3345470479-1241728991-1000 -Name property -Value value
```

Key	Value
---	----
property	value

Add metadata with a name of 'property' and a value of 'value' to the Administrator with the identifier 'S-1-5-21-1505241163-3345470479-1241728991-1000'.

Set-AdminDBConnection

May 04, 2016
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

May 04, 2016
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

May 04, 2016
Adds or updates metadata on the given Role.

Syntax

```
Set-AdminRoleMetadata [-RoleId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Set-AdminRoleMetadata [-RoleId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Set-AdminRoleMetadata [-RoleName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Set-AdminRoleMetadata [-RoleName] <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]

Set-AdminRoleMetadata [-InputObject] <Role[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]

Set-AdminRoleMetadata [-InputObject] <Role[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]
```

Detailed Description

Provides the ability for additional custom data to be stored against given Role objects.

Related topics

[Remove-AdminRoleMetadata](#)

Parameters

-RoleId<Guid>

Id of the Role

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-RoleName<String>

Name of the Role

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Role[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the property name of the metadata to be added. The property must be unique for the Role specified. The property cannot contain any of the following characters \;#.*?=<>|[]()"

Required?	true
Default Value	
Accept Pipeline Input?	false

-Value<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

System.Collections.Generic.Dictionary[String,String]

Set-AdminRoleMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

```
c:\PS>Set-AdminRoleMetadata -RoleId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

Key	Value
---	-----
property	value

Add metadata with a name of 'property' and a value of 'value' to the Role with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

Set-AdminScope

May 04, 2016
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

May 04, 2016

Adds or updates metadata on the given Scope.

Syntax

```
Set-AdminScopeMetadata [-ScopeId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-AdminScopeMetadata [-ScopeId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-AdminScopeMetadata [-ScopeName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-AdminScopeMetadata [-ScopeName] <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Set-AdminScopeMetadata [-InputObject] <Scope[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Set-AdminScopeMetadata [-InputObject] <Scope[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

Provides the ability for additional custom data to be stored against given Scope objects.

Related topics

[Remove-AdminScopeMetadata](#)

Parameters

-ScopeId<Guid>

Id of the Scope

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-ScopeName<String>

Name of the Scope

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Scope[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the property name of the metadata to be added. The property must be unique for the Scope specified. The property cannot contain any of the following characters \;#. *?=<> | []()"

Required?	true
Default Value	
Accept Pipeline Input?	false

-Value<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

System.Collections.Generic.Dictionary[String,String]

Set-AdminScopeMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

c:\PS>Set-AdminScopeMetadata -ScopeId 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

May 04, 2016

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

May 04, 2016

Retrieves the scopes where the specified operation is permitted.

Syntax

```
Test-AdminAccess [-Operation] <String[]> [-Annotate] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

This cmdlet evaluates what rights the current user has, and from these determines the scopes where the specified operation is permitted.

Operations are the indivisible unit of functionality that each XenDesktop service can perform, and usually correspond to individual cmdlets.

If you specify the -Annotate option or specify multiple operations to check, the resulting object is annotated with the operation the result relates to.

Related topics

Parameters

-Operation<String[]>

The operation to query.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Annotate<SwitchParameter>

Annotates each result with the operation it relates to.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name

or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

Citrix.DelegatedAdmin.Sdk.ScopeReference

The list of permissible scopes for the specified single operation.PSObject

The list of permissible scopes for each operation. This type of object is returned when the -Annotate option or multiple operations are specified.

Notes

If the specified operation has unrestricted access a single object is returned representing the 'All' scope with a ScopeId of Guid.Empty (00000000-0000-0000-0000-000000000000).

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Test-AdminAccess -Operation 'Broker:GetCatalog'
```

Queries the scopes where 'Broker:GetCatalog' is permitted.

----- **EXAMPLE 2** -----

```
C:\PS> Test-AdminAccess -Operation 'Broker:GetCatalog','Broker:GetMachine'
```

Queries the scopes where 'Broker:GetCatalog' or 'Broker:GetMachine' are permitted.

Test-AdminDBConnection

May 04, 2016
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

Nov 02, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016

Removes metadata from the given Service.

Syntax

```
Remove-EnvTestServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-EnvTestServiceMetadata [-ServiceHostId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-EnvTestServiceMetadata [-InputObject] <Service[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-EnvTestServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

Provides the ability to remove metadata from the given Service.

Related topics

[Set-EnvTestServiceMetadata](#)

Parameters

-ServiceHostId<Guid>

Id of the Service

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Service[]>

Objects to which metadata is to be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false

Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-EnvTestService | % { Remove-EnvTestServiceMetadata -Map $_.MetadataMap }
```

Remove all metadata from all Service objects.

Remove-EnvTestTask

May 04, 2016

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

May 04, 2016

Removes metadata from the given Task.

Syntax

```
Remove-EnvTestTaskMetadata [-TaskTaskId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-EnvTestTaskMetadata [-TaskTaskId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-EnvTestTaskMetadata [-InputObject] <EnvTestTask[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Remove-EnvTestTaskMetadata [-InputObject] <EnvTestTask[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

Detailed Description

Provides the ability to remove metadata from the given Task.

Related topics

[Set-EnvTestTaskMetadata](#)

Parameters

-TaskTaskId<Guid>

Id of the Task

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<EnvTestTask[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false

Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-EnvTestTask | % { Remove-EnvTestTaskMetadata -Map $_.MetadataMap }
```

Remove all metadata from all Task objects.

Reset-EnvTestServiceGroupMembership

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016

Adds or updates metadata on the given Service.

Syntax

```
Set-EnvTestServiceMetadata [-ServiceHostId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-EnvTestServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-EnvTestServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-EnvTestServiceMetadata [-InputObject] <Service[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Allows you to store additional custom data against given Service objects.

Related topics

[Remove-EnvTestServiceMetadata](#)

Parameters

-ServiceHostId<Guid>

Id of the Service

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Service[]>

Objects to which metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the property name of the metadata to be added. The property must be unique for the Service specified. The property cannot contain any of the following characters \/:;#.*?=<>|[]()''

Required?	true
Default Value	
Accept Pipeline Input?	false

-Value<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

System.Collections.Generic.Dictionary[String,String]

Set-EnvTestServiceMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

c:\PS>Set-EnvTestServiceMetadata -ServiceHostId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value

Key	Value
---	----
property	value

Add metadata with a name of 'property' and a value of 'value' to the Service with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

Set-EnvTestTaskMetadata

May 04, 2016
Adds or updates metadata on the given Task.

Syntax

Set-EnvTestTaskMetadata [-TaskTaskId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Set-EnvTestTaskMetadata [-TaskTaskId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Set-EnvTestTaskMetadata [-InputObject] <EnvTestTask[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Set-EnvTestTaskMetadata [-InputObject] <EnvTestTask[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Detailed Description

Provides the ability for additional custom data to be stored against given Task objects.

Related topics

[Remove-EnvTestTaskMetadata](#)

Parameters

-TaskTaskId<Guid>

Id of the Task

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<EnvTestTask[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the property name of the metadata to be added. The property must be unique for the Task specified. The property cannot contain any of the following characters \/:;#.*?=<>|[]()"

Required?	true
Default Value	
Accept Pipeline Input?	false

-Value<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

System.Collections.Generic.Dictionary[String,String]

Set-EnvTestTaskMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

```
c:\PS>Set-EnvTestTaskMetadata -TaskTaskId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

Key	Value
---	-----
property	value

Add metadata with a name of 'property' and a value of 'value' to the Task with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

Start-EnvTestTask

May 04, 2016
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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
Adds metadata on the given HostingUnit.

Syntax

```
Add-HypHostingUnitMetadata [-HostingUnitUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Add-HypHostingUnitMetadata [-HostingUnitUid] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]

Add-HypHostingUnitMetadata [-HostingUnitName] <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]

Add-HypHostingUnitMetadata [-HostingUnitName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Add-HypHostingUnitMetadata [-InputObject] <HostingUnit[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Add-HypHostingUnitMetadata [-InputObject] <HostingUnit[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]
```

Detailed Description

Provides the ability for additional custom data to be stored against given HostingUnit objects. This cmdlet will not overwrite existing metadata on an object - use the Set-HypHostingUnitMetadata cmdlet instead.

Related topics

- [Set-HypHostingUnitMetadata](#)
- [Remove-HypHostingUnitMetadata](#)

Parameters

-HostingUnitUid<Guid>

Id of the HostingUnit

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-HostingUnitName<String>

Name of the HostingUnit

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<HostingUnit[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the property name of the metadata to be added. The property must be unique for the HostingUnit specified. The property cannot contain any of the following characters \/:;#.*?=<>|[]()"

Required?	true
Default Value	
Accept Pipeline Input?	false

-Value<String>

Specifies the value for the property.

Required?	true

Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

Citrix.Host.Sdk.Metadata

Add-HypHostingUnitMetadata returns an array of objects containing the new definition of the metadata.

\n Property <string>

\n Specifies the name of the property.

\n Value <string>

\n Specifies the value for the property.

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DuplicateObject

One of the specified metadata already exists.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

```
c:\PS>Add-HypHostingUnitMetadata -HostingUnitUid 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

Property	Value
property	value

Add metadata with a name of 'property' and a value of 'value' to the HostingUnit with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

Add-HypHostingUnitNetwork

May 04, 2016
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

May 04, 2016
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

May 04, 2016
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

May 04, 2016
Adds metadata on the given HypervisorConnection.

Syntax

Add-HypHypervisorConnectionMetadata [-HypervisorConnectionUid] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Add-HypHypervisorConnectionMetadata [-HypervisorConnectionUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Add-HypHypervisorConnectionMetadata [-HypervisorConnectionName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Add-HypHypervisorConnectionMetadata [-HypervisorConnectionName] <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Add-HypHypervisorConnectionMetadata [-InputObject] <HypervisorConnection[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Add-HypHypervisorConnectionMetadata [-InputObject] <HypervisorConnection[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Detailed Description

Provides the ability for additional custom data to be stored against given HypervisorConnection objects. This cmdlet will not overwrite existing metadata on an object - use the Set-HypHypervisorConnectionMetadata cmdlet instead.

Related topics

[Set-HypHypervisorConnectionMetadata](#)

[Remove-HypHypervisorConnectionMetadata](#)

Parameters

-HypervisorConnectionUid<Guid>

Id of the HypervisorConnection

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-HypervisorConnectionName<String>

Name of the HypervisorConnection

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<HypervisorConnection[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the property name of the metadata to be added. The property must be unique for the HypervisorConnection specified. The property cannot contain any of the following characters \/:;#.*?=<>|[]0""

Required?	true
Default Value	
Accept Pipeline Input?	false

-Value<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of 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

May 04, 2016
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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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 logon (providing the schema does not already exist).

If the ScriptType parameter is set to 'Instance', the script will:

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist) and associate it with a user.

If the ScriptType parameter is set to 'Login', the script will:

Create the logon (providing the schema does not already exist) and associate it with a pre-existing user of the same name.

If the LocalDatabase parameter is included, the NetworkService account will be added to the list of accounts permitted to access the database. This is required only if the database is run on a controller.

If the command fails, the following errors can be returned.

Error Codes

GetSchemasFailed

The database schema could not be found.

ActiveDirectoryAccountResolutionFailed

The specified Active Directory account or Group could not be found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

```
c:\PS>Get-HypDBSchema -DatabaseName MyDB -ServiceGroupName MyServiceGroup > c:\HypSchema.sql
```

Get the full database schema for site data store of the Host Service and copy it to a file called 'c:\HypSchema.sql'.

This script can then be used to create the schema in a pre-existing database named 'MyDB' that does not already contain a Host Service site schema.

----- EXAMPLE 2 -----

```
c:\PS>Get-HypDBSchema -DatabaseName MyDB -scriptType Login > c:\HostLogins.sql
```

Get the logon scripts for the Host Service.

Get-HypDBVersionChangeScript

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Gets the details of the scoped objects for the Host Service.

Syntax

```
Get-HypScopedObject [-ScopeId <Guid>] [-ScopeName <String>] [-ObjectType <ScopedObjectType>] [-ObjectId <String>] [-ObjectName <String>] [-Description <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Returns a list of directly scoped objects including the names and identifiers of both the scope and object as well as the object description for display purposes.

There will be at least one result for every directly scoped object. When an object is associated with multiple scopes the output contains one result per scope duplicating the object details.

No records are returned for the All scope, though if an object is not in any scope a result with a null ScopeId and ScopeName will be returned.

Related topics

Parameters

-ScopeId<Guid>

Gets scoped object entries for the given scope identifier.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-ScopeName<String>

Gets scoped object entries with the given scope name.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-ObjectType<ScopedObjectType>

Gets scoped object entries for objects of the given type.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-ObjectId<String>

Gets scoped object entries for objects with the specified object identifier.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-ObjectName<String>

Gets scoped object entries for objects with the specified object identifier.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Description<String>

Gets scoped object entries for objects with the specified description.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about_Hyp_Filtering for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell-style filter expression. See about_Hyp_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

Citrix.Host.Sdk.ScopedObject

The Get-HypScopedObject command returns an object containing the following properties:

ScopeId <Guid?>

Specifies the unique identifier of the scope.

ScopeName <String>

Specifies the display name of the scope.

ObjectType <ScopedObjectType>

Type of the object this entry relates to.

ObjectId <String>

Unique identifier of the object.

ObjectName <String>

Display name of the object

Description <String>

Description of the object (possibly \$null if the object type does not have a description).

Notes

If the command fails, the following errors can be returned.

Error Codes

UnknownObject

One of the specified objects was not found.

PartialData

Only a subset of the available data was returned.

InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

CouldNotQueryDatabase

The query required to get the database was not defined.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

```
c:\PS>Get-HypScopedObject -ObjectType Scheme
```

ScopelId : eff6f464-f1ee-4442-add3-99982e0cec01

ScopeName : Sales

ObjectType : Scheme

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-HypService

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016
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

May 04, 2016
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

May 04, 2016

Removes metadata from the given HypervisorConnection.

Syntax

```
Remove-HypHypervisorConnectionMetadata [-HypervisorConnectionUid] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypHypervisorConnectionMetadata [-HypervisorConnectionUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypHypervisorConnectionMetadata [-HypervisorConnectionName] <String> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypHypervisorConnectionMetadata [-HypervisorConnectionName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypHypervisorConnectionMetadata [-InputObject] <HypervisorConnection[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypHypervisorConnectionMetadata [-InputObject] <HypervisorConnection[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Provides the ability to remove metadata from the given HypervisorConnection.

Related topics

[Add-HypHypervisorConnectionMetadata](#)

[Set-HypHypervisorConnectionMetadata](#)

Parameters

-HypervisorConnectionUid<Guid>

Id of the HypervisorConnection

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-HypervisorConnectionName<String>

Name of the HypervisorConnection

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<HypervisorConnection[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create

high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-HypHypervisorConnection | % { Remove-HypHypervisorConnectionMetadata -Map $_.MetadataMap }  
Remove all metadata from all HypervisorConnection objects.
```

Remove-HypHypervisorConnectionScope

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016
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 referenced 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

May 04, 2016
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

May 04, 2016

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

May 04, 2016
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

May 04, 2016
Adds or updates metadata on the given HostingUnit.

Syntax

```
Set-HypHostingUnitMetadata [-HostingUnitUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Set-HypHostingUnitMetadata [-HostingUnitUid] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]

Set-HypHostingUnitMetadata [-HostingUnitName] <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]

Set-HypHostingUnitMetadata [-HostingUnitName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Set-HypHostingUnitMetadata [-InputObject] <HostingUnit[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]

Set-HypHostingUnitMetadata [-InputObject] <HostingUnit[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]
```

Detailed Description

Provides the ability for additional custom data to be stored against given HostingUnit objects.

Related topics

- [Add-HypHostingUnitMetadata](#)
- [Remove-HypHostingUnitMetadata](#)

Parameters

-HostingUnitUid<Guid>
Id of the HostingUnit

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-HostingUnitName<String>
Name of the HostingUnit

Required?	true

Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<HostingUnit[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the property name of the metadata to be added. The property must be unique for the HostingUnit specified. The property cannot contain any of the following characters \;#. *?=<>|[]{}"

Required?	true
Default Value	
Accept Pipeline Input?	false

-Value<String>

Specifies the value for the property.

Required?	true

Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

System.Collections.Generic.Dictionary[String,String]

Set-HypHostingUnitMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

c:\PS>Set-HypHostingUnitMetadata -HostingUnitUid 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value

Key	Value
---	-----
property	value

Add metadata with a name of 'property' and a value of 'value' to the HostingUnit with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

Set-HypHostingUnitStorage

May 04, 2016
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
```

```
HostingUnitUid      : bcd3d649-86d1-4aa8-8ec2-d322b6a2c457
HostingUnitName     : MyHostingUnit
HypervisorConnection : MyConnection
RootPath            : /
RootId              :
NetworkPath         : /Network 0.network
```

```
NetworkId      : ab47080b-ca15-771a-c8dc-6ad9650158f1
Storage        : {/Local storage.storage, /newStorage.storage}
PersonalvDiskStorage : {/newStorage.storage}
VMTaggingEnabled : True
Metadata       : {}
```

The command updates a storage location called "newStorage.storage" associated with the hosting unit called "MyHostingUnit".

Set-HypHypervisorConnectionMetadata

May 04, 2016
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

May 04, 2016

Adds or updates metadata on the given Service.

Syntax

```
Set-HypServiceMetadata [-ServiceHostId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-HypServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-HypServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-HypServiceMetadata [-InputObject] <Service[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Allows you to store additional custom data against given Service objects.

Related topics

[Remove-HypServiceMetadata](#)

Parameters

-ServiceHostId<Guid>

Id of the Service

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Service[]>

Objects to which metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the property name of the metadata to be added. The property must be unique for the Service specified. The property cannot contain any of the following characters \/:#.*?=<>|[]()''

Required?	true
Default Value	
Accept Pipeline Input?	false

-Value<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

System.Collections.Generic.Dictionary[String,String]

Set-HypServiceMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

```
c:\PS>Set-HypServiceMetadata -ServiceHostId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

Key	Value
---	----
property	value

Add metadata with a name of 'property' and a value of 'value' to the Service with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

Set-HypVolumeServiceConfiguration

May 04, 2016
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

May 04, 2016
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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016
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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Gets the details of the scoped objects for the MachineCreation Service.

Syntax

```
Get-ProvScopedObject [-ScopelId <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 ScopelId and ScopeName will be returned.

Related topics

Parameters

-ScopelId<Guid>

Gets scoped object entries for the given scope identifier.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-ScopeName<String>

Gets scoped object entries with the given scope name.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-ObjectType<ScopedObjectType>

Gets scoped object entries for objects of the given type.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-ObjectId<String>

Gets scoped object entries for objects with the specified object identifier.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-ObjectName<String>

Gets scoped object entries for objects with the specified object identifier.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

-Description<String>

Gets scoped object entries for objects with the specified description.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Property<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ReturnTotalRecordCount<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about_Prov_Filtering for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

-Filter<String>

Gets records that match a PowerShell-style filter expression. See [about_Prov_Filtering](#) for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

Citrix.MachineCreation.Sdk.ScopedObject

The Get-ProvScopedObject command returns an object containing the following properties:

ScopeId <Guid?>

Specifies the unique identifier of the scope.

ScopeName <String>

Specifies the display name of the scope.

ObjectType <ScopedObjectType>

Type of the object this entry relates to.

ObjectId <String>

Unique identifier of the object.

ObjectName <String>

Display name of the object

Description <String>

Description of the object (possibly \$null if the object type does not have a description).

Notes

If the command fails, the following errors can be returned.

Error Codes

UnknownObject

One of the specified objects was not found.

PartialData

Only a subset of the available data was returned.

InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

CouldNotQueryDatabase

The query required to get the database was not defined.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ProvScopedObject -ObjectType Scheme
```

```
ScopelId   : eff6f464-f1ee-4442-add3-99982e0cec01
ScopeName  : Sales
ObjectType : Scheme
ObjectId   : cd4174ee-9e4b-4e57-b126-9dbf757fe493
ObjectName : MyExampleScheme
Description : Test scheme
```

```
ScopelId   : 304e0fa7-d390-47f0-a94f-7e956a324c41
ScopeName  : Finance
ObjectType : Scheme
ObjectId   : cd4174ee-9e4b-4e57-b126-9dbf757fe493
ObjectName : MyExampleScheme
Description : Test scheme
```

```
ScopelId   :
ScopeName  :
ObjectType : Scheme
ObjectId   : 5062e46b-71bc-4ac9-901a-30fe6797e2f6
ObjectName : AnotherScheme
Description : Another scheme in no scopes
```

Gets all of the scoped objects with type Scheme. The example output shows a scheme object (MyExampleScheme) in two scopes Sales and Finance, and another scheme (AnotherScheme) that is not in any scope. The ScopelId and ScopeName values returned are null in the final record.

Get-ProvService

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016

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

May 04, 2016

Removes metadata from the given ProvisioningScheme.

Syntax

```
Remove-ProvSchemeMetadata [-ProvisioningSchemeUid] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ProvSchemeMetadata [-ProvisioningSchemeUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ProvSchemeMetadata [-ProvisioningSchemeName] <String> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ProvSchemeMetadata [-ProvisioningSchemeName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ProvSchemeMetadata [-InputObject] <ProvisioningScheme[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ProvSchemeMetadata [-InputObject] <ProvisioningScheme[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Provides the ability to remove metadata from the given ProvisioningScheme.

Related topics

[Add-ProvSchemeMetadata](#)

[Set-ProvSchemeMetadata](#)

Parameters

-ProvisioningSchemeUid<Guid>

Id of the ProvisioningScheme

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-ProvisioningSchemeName<String>

Name of the ProvisioningScheme

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<ProvisioningScheme[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create

high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ProvScheme | Remove-ProvSchemeMetadata
Remove all metadata from all ProvisioningScheme objects.
```

Remove-ProvSchemeScope

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Removes metadata from the given Service.

Syntax

```
Remove-ProvServiceMetadata [-ServiceHostId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-ProvServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-ProvServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-ProvServiceMetadata [-InputObject] <Service[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

Provides the ability to remove metadata from the given Service.

Related topics

[Set-ProvServiceMetadata](#)

Parameters

-ServiceHostId<Guid>

Id of the Service

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Service[]>

Objects to which metadata is to be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false

Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ProvService | % { Remove-ProvServiceMetadata -Map $_.MetadataMap }
```

Remove all metadata from all Service objects.

Remove-ProvTask

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016
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

May 04, 2016

Changes the parameter values for a provisioning scheme.

Syntax

```
Set-ProvScheme [-ProvisioningSchemeName] <String> [-VMCpuCount <Int32>] [-VMMemoryMB <Int32>] [-ServiceOffering <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvScheme [-ProvisioningSchemeName] <String> -IdentityPoolUid <Guid> [-VMCpuCount <Int32>] [-VMMemoryMB <Int32>] [-ServiceOffering <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvScheme [-ProvisioningSchemeName] <String> -IdentityPoolName <String> [-VMCpuCount <Int32>] [-VMMemoryMB <Int32>] [-ServiceOffering <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvScheme -ProvisioningSchemeUid <Guid> [-VMCpuCount <Int32>] [-VMMemoryMB <Int32>] [-ServiceOffering <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvScheme -ProvisioningSchemeUid <Guid> -IdentityPoolName <String> [-VMCpuCount <Int32>] [-VMMemoryMB <Int32>] [-ServiceOffering <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvScheme -ProvisioningSchemeUid <Guid> -IdentityPoolUid <Guid> [-VMCpuCount <Int32>] [-VMMemoryMB <Int32>] [-ServiceOffering <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Provides the ability to update the parameters of an existing provisioning scheme.

This allows the following parameters to be updated:

Number of CPUs that will be used for VMs created from the provisioning scheme. Maximum amount of memory that will be used for VMs created from the provisioning scheme.

To change the name of the provisioning scheme see [Rename-ProvScheme](#).

Related topics

[New-ProvScheme](#)

[Remove-ProvScheme](#)

[Get-ProvScheme](#)

[Rename-ProvScheme](#)

Parameters

-ProvisioningSchemeName<String>

The name of the provisioning scheme to be updated.

Required?	true

Default Value	
Accept Pipeline Input?	false

-IdentityPoolUid<Guid>

The identifier of an identity pool to associate with the provisioning scheme, replacing the present one.

Required?	true
Default Value	
Accept Pipeline Input?	false

-IdentityPoolName<String>

The name of an identity pool to associate with the provisioning scheme, replacing the present one.

Required?	true
Default Value	
Accept Pipeline Input?	false

-ProvisioningSchemeUid<Guid>

The identifier of the provisioning scheme to be updated.

Required?	true
Default Value	
Accept Pipeline Input?	false

-VMCpuCount<Int32>

The number of processors that virtual machines created from the provisioning scheme should use.

Required?	false

Default Value	
Accept Pipeline Input?	false

-VMMemoryMB<Int32>

The maximum amount of memory that virtual machines created from the provisioning scheme should use.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ServiceOffering<String>

The name of the cloud service offering to use when creating machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

-PassThru<SwitchParameter>

Returns the affected record. By default, this cmdlet does not generate any output.

Required?	false
Default Value	False
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false

Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

Citrix.MachineCreation.Sdk.ProvisioningScheme

This object provides details of the provisioning scheme and contains the following information:

ProvisioningSchemeUid

ProvisioningSchemeName

The name of the provisioning scheme.

CpuCount

The number of processors that VMs will be created with when using this scheme.

MemoryMB

The maximum amount of memory that VMs will be created with when using this scheme.

MasterImageVM

The path within the hosting unit provider to the VM or snapshot of which the scheme is currently using a copy.

MasterImageVMDate

The date and time that the copy of the VM image was made for the scheme.

IdentityPoolUid

The unique identifier of the identity pool (from the ADIdentity PowerShell snap-in) that the scheme uses.

IdentityPoolName

The name of the identity pool (from the ADIdentity PowerShell snap-in) that the scheme uses.

HostingUnitUid

The unique identifier of the hosting unit (from the Hosting Unit PowerShell snap-in) that the new provisioning scheme will use.

HostingUnitName

The name of the hosting unit (from the Hosting Unit PowerShell snap-in) that the new provisioning scheme will use.

CleanOnBoot

Indicates whether the VMs created are to be reset to a clean state on each boot.

TaskId

The identifier of any current task that is running for the provisioning scheme.

Notes

In the case of failure, the following errors can result.

Error Codes

ProvisioningSchemeNotFound

The specified provisioning scheme could not be located.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

CommunicationError

An error occurred while communicating with the service.

PermissionDenied

The user does not have administrative rights to perform this operation.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

HostingUnitNotFound

The hosting unit referenced by the provisioning scheme could not be resolved

ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

C:\PS> Set-ProvScheme -ProvisioningSchemeName MyScheme -VMCpuCount 2

Updates a provisioning scheme called "MyScheme" to use two processors on the VMs that are created from the provisioning scheme.

Set-ProvSchemeMetadata

May 04, 2016
Adds or updates metadata on the given ProvisioningScheme.

Syntax

```
Set-ProvSchemeMetadata [-ProvisioningSchemeUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Set-ProvSchemeMetadata [-ProvisioningSchemeUid] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]

Set-ProvSchemeMetadata [-ProvisioningSchemeName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Set-ProvSchemeMetadata [-ProvisioningSchemeName] <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]

Set-ProvSchemeMetadata [-InputObject] <ProvisioningScheme[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Set-ProvSchemeMetadata [-InputObject] <ProvisioningScheme[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]
```

Detailed Description

Provides the ability for additional custom data to be stored against given ProvisioningScheme objects.

Related topics

[Add-ProvSchemeMetadata](#)

[Remove-ProvSchemeMetadata](#)

Parameters

-ProvisioningSchemeUid<Guid>

Id of the ProvisioningScheme

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-ProvisioningSchemeName<String>

Name of the ProvisioningScheme

Required?	true
Default Value	

Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<ProvisioningScheme[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Map<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

May 04, 2016

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

May 04, 2016

Adds or updates metadata on the given Service.

Syntax

```
Set-ProvServiceMetadata [-ServiceHostId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvServiceMetadata [-InputObject] <Service[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Allows you to store additional custom data against given Service objects.

Related topics

[Remove-ProvServiceMetadata](#)

Parameters

-ServiceHostId<Guid>

Id of the Service

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Service[]>

Objects to which metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the property name of the metadata to be added. The property must be unique for the Service specified. The property cannot contain any of the following characters \/:#.*?=<>|[]()"

Required?	true
Default Value	
Accept Pipeline Input?	false

-Value<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	

Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

System.Collections.Generic.Dictionary[String,String]

Set-ProvServiceMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-ProvServiceMetadata -ServiceHostId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

Key	Value
---	----
property	value

Add metadata with a name of 'property' and a value of 'value' to the Service with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

Set-ProvTaskMetadata

May 04, 2016

Adds or updates metadata on the given Task.

Syntax

```
Set-ProvTaskMetadata [-TaskId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-ProvTaskMetadata [-TaskId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-ProvTaskMetadata [-InputObject] <Task[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-ProvTaskMetadata [-InputObject] <Task[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

Use this cmdlet to store additional custom data against given Task objects.

Related topics

[Add-ProvTaskMetadata](#)

[Remove-ProvTaskMetadata](#)

[Get-ProvTask](#)

[Stop-ProvTask](#)

[Remove-ProvTask](#)

[Switch-ProvTask](#)

Parameters

-TaskId<Guid>

Id of the Task

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Task[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the property name of the metadata to be added. The property must be unique for the Task specified. The property cannot contain any of the following characters \ ; # . * ? = < > | [] () ""

Required?	true
Default Value	
Accept Pipeline Input?	false

-Value<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

System.Collections.Generic.Dictionary[String,String]

Set-ProvTaskMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

Notes

If the command fails, the following errors can result.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

```
c:\PS>Set-ProvTaskMetadata -TaskId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

Key	Value
---	----
property	value

Add metadata with a name of 'property' and a value of 'value' to the Task with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

Stop-ProvTask

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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 configuration settings that are 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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

Removes metadata from the given Service.

Syntax

```
Remove-MonitorServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-MonitorServiceMetadata [-ServiceHostId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-MonitorServiceMetadata [-InputObject] <Service[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-MonitorServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

Provides the ability to remove metadata from the given Service.

Related topics

[Set-MonitorServiceMetadata](#)

Parameters

-ServiceHostId<Guid>

Id of the Service

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Service[]>

Objects to which metadata is to be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false

Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-MonitorService | % { Remove-MonitorServiceMetadata -Map $_.MetadataMap }
```

Remove all metadata from all Service objects.

Reset-MonitorDataStore

May 04, 2016

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

May 04, 2016

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

May 04, 2016

设置监控服务使用的配置设置。

语法

```
Set-MonitorConfiguration [-GroomSessionsRetentionDays ] [-GroomFailuresRetentionDays ] [-GroomLoadIndexesRetentionDays ] [-GroomDeletedRetentionDays ] [-GroomSummariesRetentionDays ] [-GroomMachineHotfixLogRetentionDays ] [-GroomMinuteRetentionDays ] [-DataCollectionEnabled ] [-FullPollStartHour ] [-ResolutionPollTimeHours ] [-SyncPollTimeHours ] [-DetailedSqlOutputEnabled ] [-CollectHotfixDataEnabled ] [-GroomApplicationInstanceRetentionDays ] [-LoggingId ] [-AdminAddress ] []
```

详细说明

设置监控服务使用的配置设置。使用这些设置可修改此服务的行为。

使用此命令无需配置数据库连接。

注意

对于默认值取决于产品版本的参数（如此处所述），当版本更改时，这些参数的值将设置为新的默认值。

相关主题

[Get-MonitorConfiguration](#)

参数

-GroomSessionsRetentionDays

确定会话终止后保留会话和连接记录的天数。

是否必需？	false
默认值	对于非 Platinum 版本为 7；对于 Platinum 为 90。
是否接受管道输入？	false

-GroomFailuresRetentionDays

确定创建 MachineFailureLog 和 ConnectionFailureLog 记录后保留这些记录的天数。

是否必需？	false
默认值	对于非 Platinum 版本为 7；对于 Platinum 为 90。

是否接受管道输入?	false

-GroomLoadIndexesRetentionDays

确定创建 LoadIndex 记录后保留这些记录的天数。

是否必需?	false
默认值	对于非 Platinum 版本为 7；对于 Platinum 为 90。
是否接受管道输入?	false

-GroomDeletedRetentionDays

确定 LifecycleState 为“Deleted”的 Machine、Catalog、DesktopGroup 和 Hypervisor 条目的保留天数。此操作还会删除任何相关的 Session、Connection、Summary、Failure 或 LoadIndex 记录。

是否必需?	false
默认值	对于非 Platinum 版本为 7；对于 Platinum 为 90。
是否接受管道输入?	false

-GroomSummariesRetentionDays

确定保留 DesktopGroupSummary、FailureLogSummary 和 LoadIndexSummary 记录的天数，以天为粒度。

是否必需?	false
默认值	对于非 Platinum 版本为 7；对于 Platinum 为 90。
是否接受管道输入?	false

-GroomMachineHotfixLogRetentionDays

确定保留 Machine-Hotfix 历史记录的天数，以天为粒度。

是否必需?	false
默认值	90

是否接受管道输入?	false

-GroomMinuteRetentionDays

确定保留详细数据的天数。

是否必需?	false
默认值	3
是否接受管道输入?	false

-DataCollectionEnabled

开始/停止数据收集。停止数据收集将关闭轮询，并且不会将操作事件数据保留到数据库。

是否必需?	false
默认值	True
是否接受管道输入?	false

-FullPollStartHour

一天中全面轮询应开始的时间（小时）。

是否必需?	false
默认值	
是否接受管道输入?	false

-ResolutionPollTimeHours

解析轮询工作线程的开始时间。

是否必需?	false
默认值	
是否接受管道输入?	false

-SyncPollTimeHours

同步轮询工作线程的开始时间。

是否必需？	false
默认值	
是否接受管道输入？	false

-DetailedSqlOutputEnabled

确定是否应启用 SqlLog 以便向 CDF 跟踪发送 SQL 语句

是否必需？	false
默认值	False
是否接受管道输入？	false

-CollectHotfixDataEnabled

此设置确定是将修补程序清单数据收集起来并存储在数据库中还是将其丢弃。

是否必需？	false
默认值	
是否接受管道输入？	false

-GroomApplicationInstanceRetentionDays

是否必需？	false
默认值	
是否接受管道输入？	false

-LoggingId

指定此 cmdlet 调用所属的高级操作的标识符。Citrix Studio 和 Director 通常创建高级操作。PowerShell 脚本也可以借助 Start-LogHighLevelOperation 和 Stop-LogHighLevelOperation cmdlet 在高级操作中打包一系列 cmdlet 调用。

是否必需？	false
默认值	
是否接受管道输入？	false

-AdminAddress

指定 PowerShell 管理单元将要连接的 XenDesktop 控制器的地址。可以提供主机名或 IP 地址。

是否必需？	false
默认值	Localhost。一旦有 cmdlet 提供了某个值，此值将变为默认值。
是否接受管道输入？	false

示例

----- 示例 1 -----

C:\PS>Set-MonitorConfiguration -GroomSessionsRetentionDays 5 -GroomFailuresRetentionDays 4 ...
使用新指定的值更新站点数据库中的设置。

Set-MonitorDBConnection

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016

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

May 04, 2016
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

May 04, 2016

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

May 04, 2016

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 need to be http or https, be absolute and share the same schema and port.

Required?	true
Default Value	
Accept Pipeline Input?	false

-StorefrontUrl<Uri>

The url that will be used by Receivers to contact Storefront. Http or https absolute urls are accepted.

Required?	false
Default Value	Server name and http binding.
Accept Pipeline Input?	false

-RunAsynchronously<Boolean>

If set, the command will run asynchronously.

Required?	false
Default Value	false
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

Citrix.Storefront.Sdk.Task or Citrix.Storefront.DataModel.Cluster

Returns cluster description or a task, if ran asynchronously.

Examples

----- EXAMPLE 1 -----

New-SfCluster -FarmName "XdSiteName" -XmlServices http://farm1,http://farm2 -ServerName SfServer -RunAsynchronously \$true
Creates a new cluster on server "SfServer". The server will have a default set of services containing fully-functionally Storefront server with Authentication, Store, Receiver for Web, Desktop Appliance site and the required infrastructure. The Store service will have a farm named "XdSiteName" that will contain servers farm1 and farm2, whose will be contacted using http on default port 80.

New-SfStorefrontAddress

May 04, 2016
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

May 04, 2016

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

May 04, 2016

Removes metadata from the given Service.

Syntax

```
Remove-SfServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-SfServiceMetadata [-ServiceHostId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-SfServiceMetadata [-InputObject] <Service[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-SfServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

Provides the ability to remove metadata from the given Service.

Related topics

[Set-SfServiceMetadata](#)

Parameters

-ServiceHostId<Guid>

Id of the Service

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Service[]>

Objects to which metadata is to be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false

Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-SfService | % { Remove-SfServiceMetadata -Map $_.MetadataMap }
```

Remove all metadata from all Service objects.

Remove-SfTask

May 04, 2016

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

May 04, 2016

Removes metadata from the given Task.

Syntax

```
Remove-SfTaskMetadata [-TaskId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-SfTaskMetadata [-TaskId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-SfTaskMetadata [-InputObject] <Task[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-SfTaskMetadata [-InputObject] <Task[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

Provides the ability to remove metadata from the given Task.

Related topics

[Set-SfTaskMetadata](#)

Parameters

-TaskId<Guid>

Id of the Task

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Task[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false

Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-SfTask | % { Remove-SfTaskMetadata -Map $_.MetadataMap }
```

Remove all metadata from all Task objects.

Reset-SfServiceGroupMembership

May 04, 2016
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

May 04, 2016

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

May 04, 2016
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

May 04, 2016

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

May 04, 2016

Adds or updates metadata on the given Task.

Syntax

```
Set-SfTaskMetadata [-TaskId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-SfTaskMetadata [-TaskId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-SfTaskMetadata [-InputObject] <Task[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-SfTaskMetadata [-InputObject] <Task[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

Provides the ability for additional custom data to be stored against given Task objects.

Related topics

[Remove-SfTaskMetadata](#)

Parameters

-TaskId<Guid>

Id of the Task

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-InputObject<Task[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Map<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Name<String>

Specifies the property name of the metadata to be added. The property must be unique for the Task specified. The property cannot contain any of the following characters \ ; # . * ? = < > | [] () "

Required?	true
Default Value	
Accept Pipeline Input?	false

-Value<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	

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

May 04, 2016
Tests a database connection for the Storefront Service.

Syntax

Test-SfDBConnection [-DBConnection] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Detailed Description

Tests a connection to the database in which the Storefront Service can store its state. The service will attempt to connect to the database without affecting the current connection to the database.

You do not have to clear the connection to use this command.

Related topics

[Get-SfServiceStatus](#)

[Get-SfDBConnection](#)

[Set-SfDBConnection](#)

Parameters

-DBConnection<String>

Specifies the database connection string to be tested by the Storefront Service.

Required?	true
Default Value	
Accept Pipeline Input?	false

-LoggingId<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

-AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false

Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Test-SfDBConnection command returns an object containing the status of the Storefront Service if the connection string of the specified data store were to be set to the string being tested, together with extra diagnostics information for the specified connection string.

DBRejectedConnection

The database rejected the logon attempt from the Storefront Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the Storefront Service schema has not been added to the database.

DBNotFound

The specified database could not be located with the configured connection string.

DBNewerVersionThanService

The version of the Storefront Service currently in use is incompatible with the version of the Storefront Service schema on the database. Upgrade the Storefront Service to a more recent version.

DBOlderVersionThanService

The version of the Storefront Service schema on the database is incompatible with the version of the Storefront Service currently in use. Upgrade the database schema to a more recent version.

DBVersionChangeInProgress

A database schema upgrade is currently in progress.

OK

The Set-SfDBConnection command would succeed if it were executed with the supplied connection string.

Failed

The Storefront Service has failed.

Unknown

The status of the Storefront Service cannot be determined.

Notes

If the command fails, the following errors can be returned.

Error Codes

InvalidDBConnectionString

The database connection string has an invalid format.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Test-SfDBConnection -DBConnection "Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True"
```

OK

Tests a database connection string for the Storefront Service.

----- **EXAMPLE 2** -----

```
c:\PS>Test-SfDBConnection -DBConnection "Invalid Connection String"
```

Invalid database connection string format.

Tests an invalid database connection string for the Storefront Service.

第三方声明

May 04, 2016

XenApp 和 XenDesktop 7.6 可能包含根据以下文档中定义的条款进行许可的第三方软件：

- [XenApp 7.6 和 XenDesktop 7.6 第三方声明](#)
- [FlexNet Publisher 补充文档：软件许可证](#)