Contents

Windows 10 application management

3/19/2019 • 2 minutes to read • Edit Online

Applies to

- Windows 10

Learn about managing applications in Windows 10 and Windows 10 Mobile clients.

TOPIC	DESCRIPTION
Sideload apps in Windows 10	Requirements and instructions for side-loading LOB applications on Windows 10 and Windows 10 Mobile clients
Remove background task resource restrictions	Windows provides controls to manage which experiences may run in the background.
Enable or block Windows Mixed Reality apps in the enterprise	Learn how to enable or block Windows Mixed Reality apps.
App-V	Microsoft Application Virtualization (App-V) for Windows 10 enables organizations to deliver Win32 applications to users as virtual applications
Service Host process refactoring	Changes to Service Host grouping in Windows 10
Per User services in Windows 10	Overview of per user services and instructions for viewing and disabling them in Windows 10 and Windows 2016
Disabling System Services in Windows Server	Security guidelines for disabling services in Windows Server 2016 with Desktop Experience
Understand apps in Windows 10	Overview of the different apps included by default in Windows 10 Enterprise
Deploy app upgrades on Windows 10 Mobile	How to upgrade apps on Windows 10 Mobile
Change history for Application management	This topic lists new and updated topics in the Application management documentation for Windows 10 and Windows 10 Mobile.

Sideload LOB apps in Windows 10

5/31/2019 • 2 minutes to read • Edit Online

Applies to

- Windows 10
- Windows 10 Mobile

"Line-of-Business" (LOB) apps are present in a wide range of businesses and organizations. Organizations value these apps because they solve problems unique to each business.

When you sideload an app, you deploy a signed app package to a device. You maintain the signing, hosting, and deployment of these apps. Sideloading was also available with Windows 8 and Windows 8.1

In Windows 10, sideloading is different than in earlier versions of Windows:

- You can unlock a device for sideloading using an enterprise policy, or through **Settings**
- License keys are not required
- Devices do not have to be joined to a domain

Requirements

Here's what you'll need to have:

- Devices need to be unlocked for sideloading (unlock policy enabled)
- Certificate assigned to app
- Signed app package

And here's what you'll need to do:

- Turn on sideloading - you can push a policy with an MDM provider, or you can use **Settings**.
- Trust the app - import the security certificate to the local device.
- Install the app - use PowerShell to install the app package.

How do I sideload an app on desktop

You can sideload apps on managed or unmanaged devices.

> **IMPORTANT**
> To install an app on Windows 10, in addition to following these procedures, users can also double-click any APPX/MSIX package.

To turn on sideloading for managed devices

- Deploy an enterprise policy.

To turn on sideloading for unmanaged devices

1. Open **Settings**.

2. Click **Update & Security** > **For developers**.
3. On **Use developer features**, select **Sideload apps**.

To import the security certificate

1. Open the security certificate for the appx package, and select **Install Certificate**.
2. On the **Certificate Import Wizard**, select **Local Machine**.
3. Import the certificate to the **Trusted Root Certification Authorities** folder.

   -OR-

   You can use a runtime provisioning package to import a security certificate. For information about applying a provisioning package to a Windows 10 device, see runtime instructions on Build and apply a provisioning package.

To install the app

- From the folder with the appx package, run the PowerShell `Add-AppxPackage` command to install the appx package.

How do I sideload an app on mobile

You can sideload apps on managed or unmanaged devices.

To turn on sideloading for a managed device

- Deploy an enterprise policy.

To turn on sideloading for unmanaged devices

1. Open **Settings**.
2. Click **Update & Security** > **For developers**.
3. On **Use developer features**, select **Sideload apps**.

To import the security certificate for managed devices

1. Open the security certificate for the appx package, and select **Install Certificate**.
2. On the **Certificate Import Wizard**, select **Local Machine**.
3. Import the certificate to the **Trusted Root Certification Authorities** folder.

To import the security certificate for unmanaged devices

- You can use a runtime provisioning package to import a security certificate. For information about applying a provisioning package to a Windows 10 mobile device, see runtime instructions on Build and apply a provisioning package.

To install the app

- From an email, tap a xap, appx, or appx bundle package.

  -OR-

  With your mobile device tethered to a desktop, click a xap, appx, or appx bundle package from the files system to install the app.

Remove background task resource restrictions

5/31/2019 • 3 minutes to read • Edit Online

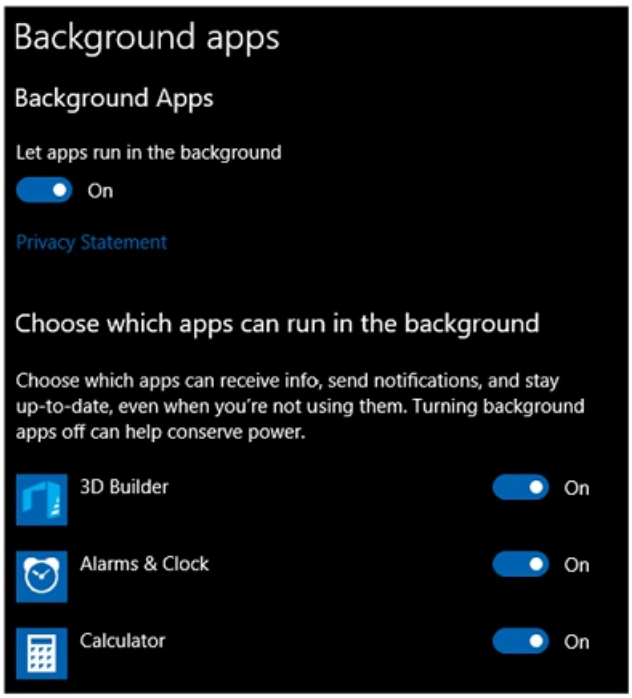
To provide the best experience for consumers, Windows provides controls that give users the choice of which experiences may run in the background.

By default, resource limits are imposed on applications. Foreground apps are given the most memory and execution time; background apps get less. Users are thus protected from poor foreground app performance and heavy battery drain.

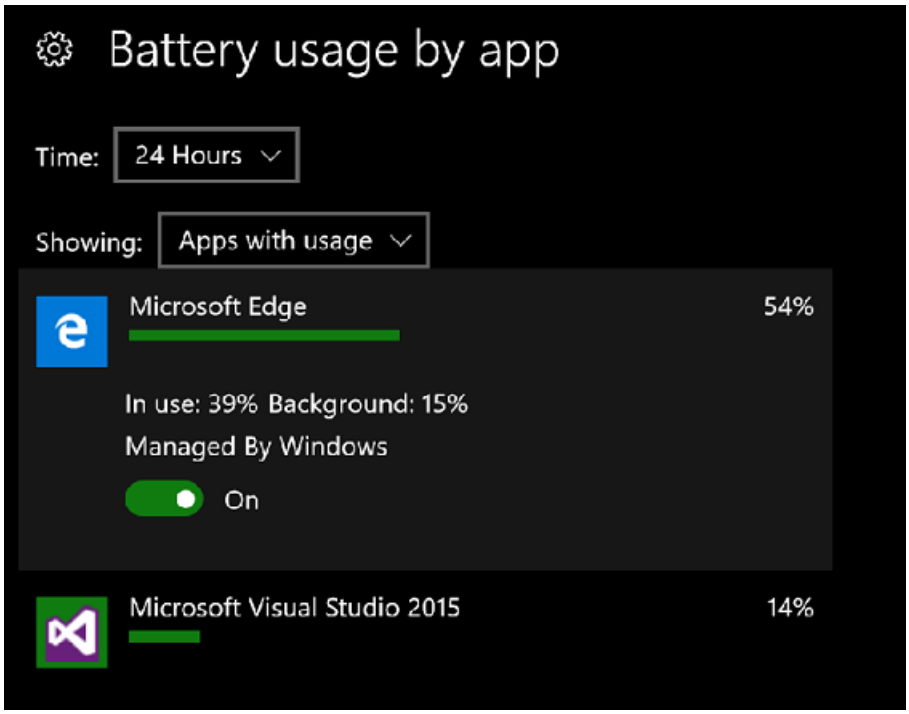
Enterprise users want the same ability to enable or limit background activity. In Windows 10, version 1703 (also known as the Creators Update), enterprises can now configure settings via policy and provisioning that control background activity.

Background activity controls

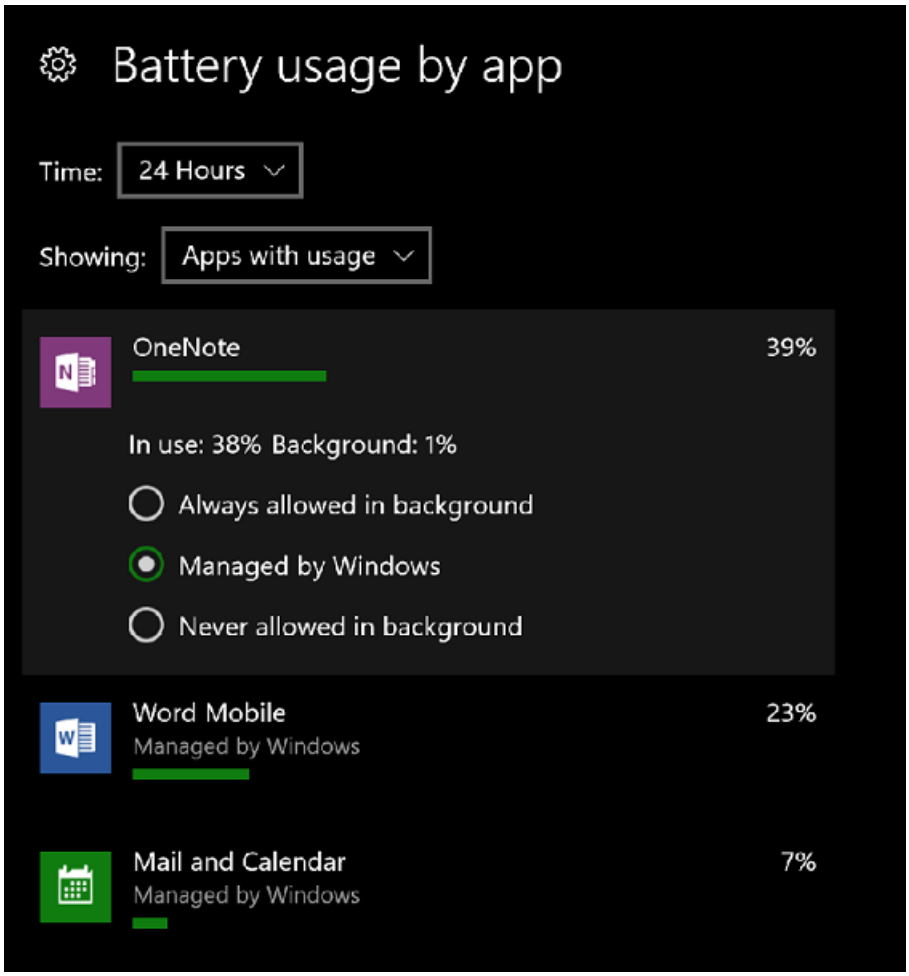
Users have the ability to control background activity for their device through two interfaces in the **Settings** app: the **Background apps** page and the **Battery usage by app** page. The **Background apps** page has a master switch to turn background activity on or off for all apps, and provides individual switches to control each app's ability to run in the background.



The **Battery usage by app** page allows fine-grained tuning of background activity. Users have the ability to set background activity to by **Managed By Windows**, as well as turning it on or off for each app. Only devices with a battery have this page available in the **Settings** app. Here is the set of available controls on desktop:



Here is the set of available controls for mobile devices:



Although the user interface differs across editions of the operating system, the policy and developer interface is consistent across Windows 10. For more information about these controls, see Optimize background activity.

Enterprise background activity controls

Starting with Windows 10, version 1703, enterprises can control background activity through mobile device management (MDM) or Group Policy. The user controls discussed above can be controlled with the following

policies:

```
./Vendor/Microsoft/Policy/Config/Privacy/LetAppsRunInBackground
./Vendor/Microsoft/Policy/Config/Privacy/LetAppsRunInBackground_ForceAllowTheseApps
./Vendor/Microsoft/Policy/Config/Privacy/LetAppsRunInBackground_ForceDenyTheseApps
./Vendor/Microsoft/Policy/Config/Privacy/LetAppsRunInBackground_UserInControlOfTheseApps
```

These policies control the background activity battery settings for Universal Windows Platform (UWP) apps. They enable apps to not be managed by the Windows system policies and not be restricted when battery saver is active. Applying these policies to a device will disable the user controls for the applications specified in the policies in the **Settings** app. See Policy CSP for more information about these policies.

An app can determine which settings are in place for itself by using BackgroundExecutionManager.RequestAccessAsync before any background activity is attempted, and then examining the returned BackgroundAccessStatus enumeration. The values of this enumeration correspond to settings in the **battery usage by App** settings page:

- **AlwaysAllowed**: Corresponds to **Always Allowed in Background** and **Managed By User**. This enables apps to run as much as possible in the background, including while the device is in battery saver mode.
- **AllowedSubjectToSystemPolicy**: This is the default value. It corresponds to **Managed by Windows**. This enables apps to run in the background as determined by Windows. If the device is currently in the battery saver state then background activities do not run.
- **DeniedDueToSystemPolicy**: Corresponds to **Managed by Windows** and indicates that the system has determined that the app cannot currently run in the background.
- **DeniedByUser**: Corresponds to **Never Allowed in the Background**. The app cannot run in the background. Either the configuration in the settings app, or enterprise policy, has defined that this app is not allowed to run in the background.

The Universal Windows Platform ensures that consumers will have great battery life and that foreground apps will perform well. Enterprises have the ability to change settings to enable scenarios specific to their business needs. Administrators can use the **Background apps** policies to enable or disable whether a UWP app can run in the background.

See also

- Run in the background indefinitely
- Policy CSP Optimize background activity

Enable or block Windows Mixed Reality apps in the enterprise

6/6/2019 • 2 minutes to read • Edit Online

Applies to

- Windows 10

Windows Mixed Reality was introduced in Windows 10, version 1709 (also known as the Fall Creators Update), as a Windows 10 Feature on Demand (FOD). Features on Demand are Windows feature packages that can be added at any time. When a Windows 10 PC needs a new feature, it can request the feature package from Windows Update.

Organizations that use Windows Server Update Services (WSUS) must take action to enable Windows Mixed Reality. Any organization that wants to prohibit use of Windows Mixed Reality can block the installation of the Mixed Reality Portal.

Enable Windows Mixed Reality in WSUS

1. Check your version of Windows 10.

   > **NOTE**
   > You must be on at least Windows 10, version 1709, to run Windows Mixed Reality.

2. Windows Mixed Reality Feature on Demand (FOD) is downloaded from Windows Update. If access to Windows Update is blocked, you must manually install the Windows Mixed Reality FOD.

   a. Download the FOD .cab file for Windows 10, version 1903, Windows 10, version 1809, Windows 10, version 1803, or Windows 10, version 1709.

   > **NOTE**
   > You must download the FOD .cab file that matches your operating system version.

   b. Use `Add-Package` to add Windows Mixed Reality FOD to the image.

   ```
   Add-Package
   Dism /Online /add-package /packagepath:(path)
   ```

   c. In **Settings** > **Update & Security** > **Windows Update**, select **Check for updates**.

IT admins can also create Side by side feature store (shared folder) to allow access to the Windows Mixed Reality FOD.

Block the Mixed Reality Portal

You can use the AppLocker configuration service provider (CSP) to block the Mixed Reality software.

In the following example, the **Id** can be any generated GUID and the **Name** can be any name you choose. Note that `BinaryName="*"` allows you to block any app executable in the Mixed Reality Portal package. **Binary/VersionRange**, as shown in the example, will block all versions of the Mixed Reality Portal app.

```
<SyncML xmlns="SYNCML:SYNCML1.2">
    <SyncBody>
        <Add>
            <CmdID>$CmdID$</CmdID>
            <Item>
                <Target>

<LocURI>./Vendor/MSFT/PolicyManager/My/ApplicationManagement/ApplicationRestrictions</LocURI>
                </Target>
                <Meta>
                    <Format xmlns="syncml:metinf">chr</Format>
                    <Type xmlns="syncml:metinf">text/plain</Type>
                </Meta>
                <Data>
                  <RuleCollection Type="Appx" EnforcementMode="Enabled">
                   <FilePublisherRule Id="a9e18c21-ff8f-43cf-b9fc-db40eed693ba" Name="(Default Rule) All
signed packaged apps" Description="Allows members of the Everyone group to run packaged apps that are signed."
UserOrGroupSid="S-1-1-0" Action="Allow">
                    <Conditions>
                      <FilePublisherCondition PublisherName="*" ProductName="*" BinaryName="*">
                        <BinaryVersionRange LowSection="0.0.0.0" HighSection="*" />
                      </FilePublisherCondition>
                    </Conditions>
                  </FilePublisherRule>
                  <FilePublisherRule Id="d26da4e7-0b01-484d-a8d3-d5b5341b2d55" Name="Block Mixed Reality
Portal" Description="" UserOrGroupSid="S-1-1-0" Action="Deny">
                   <Conditions>
                     <FilePublisherCondition PublisherName="CN=Microsoft Windows, O=Microsoft Corporation,
L=Redmond, S=Washington, C=US" ProductName="Microsoft.Windows.HolographicFirstRun" BinaryName="*">
                      <BinaryVersionRange LowSection="*" HighSection="*" />
                      </FilePublisherCondition>
                    </Conditions>
                  </FilePublisherRule>
                 </RuleCollection>>
                </Data>
            </Item>
        </Add>
        <Final/>
    </SyncBody>
</SyncML>
```

Related topics

- Mixed reality

Understand the different apps included in Windows 10

6/6/2019 • 5 minutes to read • Edit Online

Applies to: Windows 10

The following types of apps run on Windows 10:

- Windows apps - introduced in Windows 8, primarily installed from the Store app.
- Universal Windows Platform (UWP) apps - designed to work across platforms, can be installed on multiple platforms including Windows client, Windows Phone, and Xbox. All UWP apps are also Windows apps, but not all Windows apps are UWP apps.
- "Win32" apps - traditional Windows applications.

Digging into the Windows apps, there are two categories:

- Apps - All other apps, installed in C:\Program Files\WindowsApps. There are two classes of apps:
  - Provisioned: Installed in user account the first time you sign in with a new user account.
  - Installed: Installed as part of the OS.
- System apps - Apps that are installed in the C:\Windows* directory. These apps are integral to the OS.

The following tables list the system apps, installed Windows apps, and provisioned Windows apps in a standard Windows 10 Enterprise installation. (If you have a custom image, your specific apps might differ.) The tables list the app, the full name, show the app's status in Windows 10 version 1709, 1803, and 1809 and indicate whether an app can be uninstalled through the UI.

Some of the apps show up in multiple tables - that's because their status changed between versions. Make sure to check the version column for the version you are currently running.

Provisioned Windows apps

Here are the provisioned Windows apps in Windows 10 versions 1703, 1709, 1803 and 1809.

TIP

You can list all provisioned Windows apps with this PowerShell command:

```
Get-AppxProvisionedPackage -Online | Format-Table DisplayName, PackageName
```

PACKAGE NAME	APP NAME	1703	1709	1803	1809	UNINSTALL THROUGH UI?
Microsoft.3DBuilder	3D Builder	x				Yes
Microsoft.BingWeather	MSN Weather	x	x	x	x	Yes

PACKAGE NAME	APP NAME	1703	1709	1803	1809	UNINSTALL THROUGH UI?
Microsoft.DesktopAppInstaller	App Installer	x	x	x	x	Via Settings App
Microsoft.GetHelp	Get Help		x	x	x	No
Microsoft.Getstarted	Microsoft Tips	x	x	x	x	No
Microsoft.HEIFImageExtension	HEIF Image Extensions				x	No
Microsoft.Messaging	Microsoft Messaging	x	x	x	x	No
Microsoft.Microsoft3DViewer	Mixed Reality Viewer	x	x	x	x	No
Microsoft.MicrosoftOfficeHub	My Office	x	x	x	x	Yes
Microsoft.MicrosoftSolitaireCollection	Microsoft Solitaire Collection	x	x	x	x	Yes
Microsoft.MicrosoftStickyNotes	Microsoft Sticky Notes	x	x	x	x	No
Microsoft.MixedReality.Portal	Mixed Reality Portal				x	No
Microsoft.MSPaint	Paint 3D	x	x	x	x	No
Microsoft.Office.OneNote	OneNote	x	x	x	x	Yes
Microsoft.OneConnect	Paid Wi-Fi & Cellular	x	x	x	x	No
Microsoft.People	Microsoft People	x	x	x	x	No
Microsoft.Print3D	Print 3D		x	x	x	No
Microsoft.ScreenSketch	Snip & Sketch				x	No

PACKAGE NAME	APP NAME	1703	1709	1803	1809	UNINSTALL THROUGH UI?
Microsoft.SkypeApp	Skype	x	x	x	x	No
Microsoft.StorePurchaseApp	Store Purchase App	x	x	x	x	No
Microsoft.VP9VideoExtensions					x	No
Microsoft.Wallet	Microsoft Pay	x	x	x	x	No
Microsoft.WebMediaExtensions	Web Media Extensions			x	x	No
Microsoft.WebpImageExtension	Webp Image Extension				x	No
Microsoft.Windows.Photos	Microsoft Photos	x	x	x	x	No
Microsoft.WindowsAlarms	Windows Alarms & Clock	x	x	x	x	No
Microsoft.WindowsCalculator	Windows Calculator	x	x	x	x	No
Microsoft.WindowsCamera	Windows Camera	x	x	x	x	No
microsoft.windowscommunicationsapps	Mail and Calendar	x	x	x	x	No
Microsoft.WindowsFeedbackHub	Feedback Hub	x	x	x	x	No
Microsoft.WindowsMaps	Windows Maps	x	x	x	x	No
Microsoft.WindowsSoundRecorder	Windows Voice Recorder	x	x	x	x	No
Microsoft.WindowsStore	Microsoft Store	x	x	x	x	No

PACKAGE NAME	APP NAME	1703	1709	1803	1809	UNINSTALL THROUGH UI?
Microsoft.Xbox.TCUI	Xbox TCUI		x	x	x	No
Microsoft.XboxApp	Xbox	x	x	x	x	No
Microsoft.XboxGameOverlay	Xbox Game Bar	x	x	x	x	No
Microsoft.XboxGamingOverlay	Xbox Gaming Overlay			x	x	No
Microsoft.XboxIdentityProvider	Xbox Identity Provider	x	x	x	x	No
Microsoft.XboxSpeechToTextOverlay		x	x	x	x	No
Microsoft.YourPhone	Your Phone				x	No
Microsoft.ZuneMusic	Groove Music	x	x	x	x	No
Microsoft.ZuneVideo	Movies & TV	x	x	x	x	No

NOTE

The Store app can't be removed. If you want to remove and reinstall the Store app, you can only bring Store back by either restoring your system from a backup or resetting your system. Instead of removing the Store app, you should use group policies to hide or disable it.

System apps

System apps are integral to the operating system. Here are the typical system apps in Windows 10 versions 1709, 1803, and 1809.

TIP

You can list all system apps with this PowerShell command:

```
Get-AppxPackage -PackageTypeFilter Main | ? { $_.SignatureKind -eq "System" } | Sort Name | Format-Table
Name, InstallLocation
```

NAME	PACKAGE NAME	1709	1803	1809	UNINSTALL THROUGH UI?
File Picker	1527c705-839a-4832-9118-54d4Bd6a0c89		x	x	No
File Explorer	c5e2524a-ea46-4f67-841f-6a9465d9d515		x	x	No
App Resolver UX	E2A4F912-2574-4A75-9BB0-0D023378592B		x	x	No
Add Suggested Folders To Library	F46D4000-FD22-4DB4-AC8E-4E1DDDE828FE		x	x	No
	InputApp	x	x	x	No
Microsoft.AAD.Broker.Plugin	Microsoft.AAD.Broker.Plugin	x	x	x	No
Microsoft.AccountsControl	Microsoft.AccountsControl	x	x	x	No
Microsoft.AsyncTextService	Microsoft.AsyncTextService		x	x	No
Hello setup UI	Microsoft.BioEnrollment	x	x	x	No
	Microsoft.CredDialogHost	x	x	x	No
	Microsoft.ECApp	x	x	x	No
	Microsoft.LockApp	x	x	x	No
Microsoft Edge	Microsoft.MicrosoftEdge	x	x	x	No
	Microsoft.MicrosoftEdgeDevToolsClient		x	x	No
	Microsoft.PPIProjection	x	x	x	No
	Microsoft.Win32WebViewHost		x	x	No

NAME	PACKAGE NAME	1709	1803	1809	UNINSTALL THROUGH UI?
	Microsoft.Windows.Apprep.ChxApp	x	x	x	No
	Microsoft.Windows.AssignedAccessLockApp	x	x	x	No
	Microsoft.Windows.CapturePicker		x	x	No
	Microsoft.Windows.CloudExperienceHost	x	x	x	No
	Microsoft.Windows.ContentDeliveryManager	x	x	x	No
Cortana	Microsoft.Windows.Cortana	x	x	x	No
	Microsoft.Windows.Holographic.FirstRun	x	x		No
	Microsoft.Windows.OOBENetworkCaptivePort	x	x	x	No
	Microsoft.Windows.OOBENetworkConnectionFlow	x	x	x	No
	Microsoft.Windows.ParentalControls	x	x	x	No
People Hub	Microsoft.Windows.PeopleExperienceHost	x	x	x	No
	Microsoft.Windows.PinningConfirmationDialog	x	x	x	No
	Microsoft.Windows.SecHealthUI	x	x	x	No
	Microsoft.Windows.SecondaryTileExperience	x			No

NAME	PACKAGE NAME	1709	1803	1809	UNINSTALL THROUGH UI?
	Microsoft.Windows.SecureAssessmentBrowser	x	x	x	No
Start	Microsoft.Windows.ShellExperienceHost	x	x	x	No
Windows Feedback	Microsoft.WindowsFeedback	*			No
	Microsoft.XboxGameCallableUI	x	x	x	No
	Windows.CBSPreview		x	x	No
Contact Support*	Windows.ContactSupport	*			Via Settings App
Settings	Windows.immersivecontrolpanel	x	x	x	No
Print 3D	Windows.Print3D		x	x	Yes
Print UI	Windows.PrintDialog	x	x	x	No

NOTE

- The Contact Support app changed to Get Help in version 1709. Get Help is a provisioned app (instead of system app like Contact Support).

Installed Windows apps

Here are the typical installed Windows apps in Windows 10 versions 1709, 1803, and 1809.

NAME	FULL NAME	1709	1803	1809	UNINSTALL THROUGH UI?
Remote Desktop	Microsoft.RemoteDesktop	x		x	Yes
Code Writer	ActiproSoftwareLLC.562882FEEB491	x	x		Yes
Eclipse Manager	46928bounde.EclipseManager	x	x		Yes
Pandora	PandoraMediaInc.29680B314EFC2	x	x		Yes

NAME	FULL NAME	1709	1803	1809	UNINSTALL THROUGH UI?
Photoshop Express	AdobeSystemIncorporated.AdobePhotoshop	x	x		Yes
Duolingo	D5EA27B7.Duolingo-LearnLanguagesforFree	x	x		Yes
Network Speed Test	Microsoft.NetworkSpeedTest	x	x	x	Yes
News	Microsoft.BingNews	x	x	x	Yes
Sway	Microsoft.Office.Sway	x	x	x	Yes
Microsoft.Advertising	Microsoft.Advertising.Xaml	x	x	x	Yes
	Microsoft.NET.Native.Framework.1.2	x	x		Yes
	Microsoft.NET.Native.Framework.1.3	x	x		Yes
	Microsoft.NET.Native.Framework.1.6	x	x	x	Yes
	Microsoft.NET.Native.Framework.1.7		x	x	Yes
	Microsoft.NET.Native.Framework.2.0	x	x		Yes
	Microsoft.NET.Native.Runtime.1.1	x	x		Yes
	Microsoft.NET.Native.Runtime.1.3	x			Yes
	Microsoft.NET.Native.Runtime.1.4	x	x		Yes
	Microsoft.NET.Native.Runtime.1.6	x	x	x	Yes

NAME	FULL NAME	1709	1803	1809	UNINSTALL THROUGH UI?
	Microsoft.NET.Native.Runtime.1.7	x	x	x	Yes
	Microsoft.NET.Native.Runtime.2.0	x	x		Yes
	Microsoft.Services.Store.Engagement	x	x		Yes
	Microsoft.VCLibs.120.00	x	x		Yes
	Microsoft.VCLibs.140.00	x	x	x	Yes
	Microsoft.VCLibs.120.00.Universal	x			Yes
	Microsoft.VCLibs.140.00.UWPDesktop		x		Yes

How to add apps and features to Windows 10

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10

Windows 10 includes a range of applications, from system apps that support the operating system (like Settings) to "provisioned" apps (like Feedback Hub) that are installed the first time you run Windows. We also provide additional apps and features, called Features on Demand (like language packs or handwriting recognition), that you can install at any time. If you're working in a managed environment (like at work, where you have an administrator who manages your systems and resources), your admin can use Windows Update to install Features on Demand. If you're working on your own device, you can add apps and features from the Settings app.

Here's how you do that:

1. In the Search bar, search for "apps."
2. Select **Apps and features** in the results.
3. Select **Manage optional features**, and then select **Add a feature**.
4. Select the feature you want to add, like **XPS Viewer**, and then select **Install.**

And that's it. You can see the apps you have installed on the **Apps & features** page and the features on **Manage optional features**.

You can manage and uninstall apps and features from the same Settings page. Just select the app or feature, and then select **Uninstall**.

Repackage existing win32 applications to the MSIX format

5/31/2019 • 2 minutes to read • Edit Online

MSIX is a packaging format built to be safe, secure and reliable, based on a combination of .msi, .appx, App-V and ClickOnce installation technologies. You can use the MSIX packaging tool to repackage your existing Win32 applications to the MSIX format.

You can either run your installer interactively (through the UI) or create a package from the command line. Either way, you can convert an application without having the source code. Then, you can make your app available through the Microsoft Store.

- Package your favorite application installer interactively (msi, exe, App-V 5.x and ClickOnce) in MSIX format.
- Create a modification package to update an existing MSIX package.
- Bundle multiple MSIX packages for distribution.

Installing the MSIX Packaging Tool

Prerequisites

- Windows 10, version 1809 (or later)
- Participation in the Windows Insider Program (if you're using an Insider build)
- A valid Microsoft account (MSA) alias to access the app from the Microsoft Store
- Admin privileges on your PC account

Get the app from the Microsoft Store

1. Use the MSA login associated with your Windows Insider Program credentials in the Microsoft Store.
2. Open the product description page.
3. Click the install icon to begin installation.

Application Virtualization (App-V) for Windows 10 overview

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

The topics in this section provide information and instructions to help you administer App-V and its components. This information is for system administrators who manage large installations with many servers and clients, and for support personnel who interact directly with the computers or users.

Getting started with App-V

- What's new in App-V
- Evaluating App-V
- High-level architecture for App-V

Planning for App-V

- Preparing your environment for App-V
- App-V prerequisites
- Planning to deploy App-V
- App-V supported configurations
- App-V planning checklist

Deploying App-V

- Deploying the App-V Sequencer and configuring the client
- Deploying the App-V Server
- App-V deployment checklist
- Deploying Microsoft Office 2016 by using App-V
- Deploying Microsoft Office 2013 by using App-V
- Deploying Microsoft Office 2010 by using App-V

Operations for App-V

- Creating and managing App-V virtualized applications
- Automatically provision your sequencing environment using Microsoft Application Virtualization Sequencer (App-V Sequencer)
- Automatically sequence multiple apps at the same time using Microsoft Application Virtualization Sequencer (App-V Sequencer)
- Automatically update multiple apps at the same time using Microsoft Application Virtualization Sequencer (App-V Sequencer)
- Administering App-V Virtual Applications by using the Management Console
- Managing connection groups
- Deploying App-V packages by using Electronic Software Distribution (ESD)
- Using the App-V Client Management Console
- Automatically clean up unpublished packages on the App-V client
- Migrating to App-V from a previous version

- Maintaining App-V
- Administering App-V by using Windows PowerShell

Troubleshooting App-V

Technical reference for App-V

- Performance guidance for Application Virtualization
- Application publishing and client interaction
- Viewing App-V Server publishing metadata
- Running a locally installed application inside a virtual environment with virtualized applications

Getting started with App-V for Windows 10

5/31/2019 • 3 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Microsoft Application Virtualization (App-V) for Windows 10 delivers Win32 applications to users as virtual applications. Virtual applications are installed on centrally managed servers and delivered to users as a service in real time and on an as-needed basis. Users launch virtual applications from familiar access points and interact with them as if they were installed locally.

With the release of Windows 10, version 1607, App-V is included with the Windows 10 for Enterprise edition. If you're new to Windows 10 and App-V, you'll need to download, activate, and install server- and client-side components to start delivering virtual applications to users. To learn what you need to know before getting started with App-V, see the Application Virtualization (App-V) overview.

If you're already using App-V, performing an in-place upgrade to Windows 10 on user devices automatically installs the App-V client and migrates users' App-V applications and settings. For more information about how to configure an existing App-V installation after upgrading user devices to Windows 10, see Upgrading to App-V for Windows 10 from an existing installation.

> **IMPORTANT**
>
> You can upgrade your existing App-V installation to App-V for Windows from App-V versions 5.0 SP2 and higher only. If you are using an earlier version of App-V, you'll need to upgrade your existing App-V installation to App-V 5.0 SP2 before upgrading to App-V for Windows.

To learn more about previous versions of App-V, see MDOP information experience.

Getting started with App-V for Windows 10 (new installations)

To start using App-V to deliver virtual applications to users, you'll need to download, enable, and install server- and client-side components. The following table describes the App-V for Windows 10 components, what they do, and where to find them.

COMPONENT	WHAT IT DOES	WHERE TO FIND IT

COMPONENT	WHAT IT DOES	WHERE TO FIND IT
App-V server components	App-V offers five server components that work together to allow you to host and publish virtual applications, generate usage reports, and manage your App-V environment. For more details, see Deploying the App-V Server. If you're already using App-V 5.x, you don't need to redeploy the App-V server components, as they haven't changed since App-V 5.0's release.	The App-V server components are included in the Microsoft Desktop Optimization Pack (MDOP) 2015 ISO package that can be downloaded from the following locations: If you have a Microsoft Developer Network (MSDN) subscription, use the MSDN (Microsoft Developer Network) subscriptions site to download the MDOP ISO package. If you're using Windows 10 for Enterprise or Education, download it from the Volume Licensing Service Center. See Deploying the App-V Server for more information about installing and using the server components.
App-V client and App-V Remote Desktop Services (RDS) client	The App-V client is the component that runs virtualized applications on user devices, allowing users to interact with icons and file names to start virtualized applications.	The App-V client is automatically installed with Windows 10, version 1607. To learn how to enable the client, see Enable the App-V desktop client.
App-V sequencer	Use the App-V sequencer to convert Win32 applications into virtual packages for deployment to user devices. Devices must run the App-V client to allow users to interact with virtual applications.	Installed with the Windows Assessment and Deployment kit (ADK) for Windows 10, version 1607.

For more information about these components, see High Level Architecture for App-V.

If you're new to App-V, it's a good idea to read the documentation thoroughly. Before deploying App-V in a production environment, you can ensure installation goes smoothly by validating your deployment plan in a test network environment. You might also consider taking a class about relevant technologies. To get started, see the Microsoft Training Overview.

Getting started with App-V

What's new in App-V provides a high-level overview of App-V and how it can be used in your organization.

Evaluating App-V provides information about how you can best evaluate App-V for use in your organization.

High Level Architecture for App-V provides a description of the App-V features and how they work together.

Other resources for this product

- Application Virtualization (App-V) overview
- Planning for App-V
- Deploying App-V
- Operations for App-V
- Troubleshooting App-V

- Technical reference for App-V

What's new in App-V for Windows 10, version 1703 and earlier

6/10/2019 • 3 minutes to read • Edit Online

Applies to: Windows 10, version 1703 and earlier

Microsoft Application Virtualization (App-V) for Windows 10 delivers Win32 applications to users as virtual applications. Virtual applications are installed on centrally managed servers and delivered to users as a service in real time and on an as-needed basis. Users launch virtual applications from familiar access points and interact with them as if they were installed locally.

What's new in App-V Windows 10, version 1703

Auto-sequence and update your App-V packages singly or as a batch

Previous versions of the App-V Sequencer required manual sequencing and updating of app packages. This was time-consuming and required extensive interaction, causing many companies to deploy brand-new packages rather than update an existing one. Windows 10, version 1703 introduces the App-V Auto-Sequencer, which automatically sequences your app packages, improving your overall experience by streamlining prerequisite environment provisioning, automating app installation, and expediting package updating setup.

Using the automatic sequencer to package your apps gives you the following benefits:

- **Automatic virtual machine (VM) sequencing environment provisioning**. To learn more, see Automatically provision your sequencing environment using Microsoft Application Virtualization Sequencer (App-V Sequencer).
- **Package batch-sequencing**. This means that multiple apps can be sequenced at the same time, in a single group. To learn more, see Automatically sequence multiple apps at the same time using Microsoft Application Virtualization Sequencer (App-V Sequencer).
- **Package batch-updating**. This means that multiple apps can be updated at the same time, in a single group. To learn more, see Automatically update multiple apps at the same time using Microsoft Application Virtualization Sequencer (App-V Sequencer).

Updates to the App-V project template

Starting with Windows 10, version 1703, you can now save an App-V project template (.appvt) file as part of a sequenced App-V package. This file will automatically load every time you open the package for edits or updates. Your template can include general option settings, file exclusion list settings, and target operating system settings. To learn more, see Create and apply an App-V project template to a sequenced App-V package.

Automatically clean up unpublished App-V packages from the App-V client

Previous versions of App-V have required you to manually remove your unpublished packages from your client devices, to free up additional storage space. Windows 10, version 1703 introduces the ability to use PowerShell or Group Policy settings to automatically clean up your unpublished packages after a device restart. To learn more, see Automatically clean up unpublished packages on the App-V client.

What's new in App-V in Windows 10, version 1607

App-V is now a feature in Windows 10

With Windows 10, version 1607 and later releases, App-V is now included with Windows 10 for Enterprise and

Windows 10 for Education and is no longer part of the Microsoft Desktop Optimization Pack.

To learn more about earlier versions of App-V, see MDOP Information Experience.

The changes in App-V for Windows 10, version 1607 impact existing implementations of App-V in the following ways:

- The App-V client is installed on user devices automatically with Windows 10, version 1607, and no longer has to be deployed separately. Performing an in-place upgrade to Windows 10, version 1607, on user devices automatically installs the App-V client.
- In previous releases of App-V, the application sequencer was included in the Microsoft Desktop Optimization Pack. Although you'll need to use the new application sequencer to create new virtualized applications, existing virtualized applications will continue to work. The App-V application sequencer is available from the Windows 10 Assessment and Deployment Kit (ADK).

> **NOTE**
> If you're already using App-V 5.x, you don't need to redeploy the App-V server components, as they haven't changed since App-V 5.0's release.

For more information about how to configure an existing App-V installation after upgrading user devices to Windows 10, see Upgrading to App-V for Windows 10 from an existing installation and Migrating to App-V for Windows 10 from a previous version.

> **IMPORTANT**
> You can only upgrade your existing App-V installation to Windows 10, version 1607 if it's version 5.0 SP2 or higher. If you're using an older version of App-V, you'll need to upgrade from that version to App-V 5.0 SP2 before you can upgrade to Windows 10, version 1607.

Support for System Center

App-V supports System Center 2016 and System Center 2012 R2 Configuration Manager SP1. See Planning for App-V Integration with Configuration Manager to learn more about how to integrate your App-V environment with Configuration Manager.

Related topics

- Release Notes for App-V for Windows 10, version 1607
- Release Notes for App-V for Windows 10, version 1703

Release Notes for App-V for Windows 10, version 1607

6/10/2019 • 7 minutes to read • Edit Online

Applies to

- Windows 10, version 1607

The following are known issues and workarounds for Application Virtualization (App-V) running on Windows 10, version 1607.

Windows Installer packages (.msi files) generated by the App-V sequencer (version 5.1 and earlier) fail to install on computers with the in-box App-V client

MSI packages that were generated using an App-V sequencer from previous versions of App-V (App-V versions 5.1 and earlier) include a check to validate that the App-V client is installed on client devices before allowing the MSI package to install. Now that the App-V client is installed automatically when you upgrade user devices to Windows 10, version 1607, the pre-requisite check fails and causes the MSI to fail.

Workaround:

1. Install the latest App-V sequencer, which you can get from the Windows Assessment and Deployment Kit (ADK) for Windows 10, version 1607. See Download the Windows ADK. For more information, see Install the App-V Sequencer.
2. Ensure that you have installed the **MSI Tools** included in the Windows 10 SDK, available as follows:
   - For the **Visual Studio Community 2015 with Update 3** client, which includes the latest Windows 10 SDK and developer tools, see Downloads and tools for Windows 10.
   - For the standalone Windows 10 SDK without other tools, see Standalone Windows 10 SDK.
3. Copy msidb.exe from the default path of the Windows SDK installation (**C:\Program Files (x86)\Windows Kits\10**) to a different directory. For example: **C:\MyMsiTools\bin**
4. From an elevated Windows PowerShell prompt, navigate to the following folder:

   <Windows Kits 10 installation folder>**\Microsoft Application Virtualization\Sequencer**

   By default, this path will be:
   C:\Program Files (x86)\Windows Kits\10\Microsoft Application Virtualization\Sequencer
5. Run the following command:

   ```
   Update-AppvPackageMsi -MsiPackage "<path to App-V Package .msi file>" -MsSdkPath "<path>"
   ```

   where the path is to the new directory (**C:\MyMsiTools\ for this example**).

Error occurs during publishing refresh between App-V 5.0 SP3 Management Server and App-V Client on Windows 10

An error is generated during publishing refresh when synchronizing packages from the App-V 5.0 SP3

management server to an App-V client on Windows 10. This error occurs because the App-V 5.0 SP3 server does not understand the Windows 10 operating system that is specified in the publishing URL. The issue is fixed for App-V publishing server, but is not backported to versions of App-V 5.0 SP3 or earlier.

Workaround: Upgrade the App-V 5.0 Management server to the App-V Management server for Windows 10 Clients.

Custom configurations do not get applied for packages that will be published globally if they are set using the App-V Server

If you assign a package to an AD group that contains machine accounts and apply a custom configuration to that group using the App-V Server, the custom configuration will not be applied to those machines. The App-V Client will publish packages assigned to a machine account globally. However, it stores custom configuration files per user in each user's profile. Globally published packages will not have access to this custom configuration.

Workaround: Do one of the following:

- Assign the package to groups containing only user accounts. This will ensure that the package's custom configuration will be stored in each user's profile and will be applied correctly.
- Create a custom deployment configuration file and apply it to the package on the client using the Add-AppvClientPackage cmdlet with the –DynamicDeploymentConfiguration parameter. See About App-V Dynamic Configuration for more information.
- Create a new package with the custom configuration using the App-V Sequencer.

Server files not deleted after new App-V Server installation

If you uninstall the App-V 5.0 SP1 Server and then install the App-V Server, the installation fails, the wrong version of the Management server is installed, and an error message is returned. The issue occurs because the Server files are not being deleted when you uninstall App-V 5.0 SP1, so the installation process does an upgrade instead of a new installation.

Workaround: Delete this registry key before you start installing App-V:

Under HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall, locate and delete the installation GUID key that contains the DWORD value "DisplayName" with value data "Microsoft Application Virtualization (App-V) Server". This is the only key that should be deleted.

File type associations added manually are not saved correctly

File type associations added to an application package manually using the Shortcuts and FTAs tab at the end of the application upgrade wizard are not saved correctly. They will not be available to the App-V Client or to the Sequencer when updating the saved package again.

Workaround: To add a file type association, open the package for modification and run the update wizard. During the Installation step, add the new file type association through the operating system. The sequencer will detect the new association in the system registry and add it to the package's virtual registry, where it will be available to the client.

When streaming packages in Shared Content Store (SCS) mode to a client that is also managed with AppLocker, additional data is written to the local disk.

To decrease the amount of data written to a client's local disk, you can enable SCS mode on the App-V Client to

stream the contents of a package on demand. However, if AppLocker manages an application within the package, some data might be written to the client's local disk that would not otherwise be written.

Workaround: None

In the Management Console Add Package dialog box, the Browse button is not available when using Chrome or Firefox

On the Packages page of the Management Console, if you click **Add or Upgrade** in the lower-right corner, the **Add Package** dialog box appears. If you are accessing the Management Console using Chrome or Firefox as your browser, you will not be able to browse to the location of the package.

Workaround: Type or copy and paste the path to the package into the **Add Package** input field. If the Management Console has access to this path, you will be able to add the package. If the package is on a network share, you can browse to the location using File Explorer by doing these steps:

1. While pressing **Shift**, right-click on the package file
2. Select **Copy as path**
3. Paste the path into the **Add Package** dialog box input field

Upgrading App-V Management Server to 5.1 sometimes fails with the message "A database error occurred"

If you install the App-V 5.0 SP1 Management Server, and then try to upgrade to App-V Server when multiple connection groups are configured and enabled, the following error is displayed: "A database error occurred. Reason: 'Invalid column name 'PackageOptional'. Invalid column name 'VersionOptional'."

Workaround: Run this command on your SQL database:

```
ALTER TABLE AppVManagement.dbo.PackageGroupMembers ADD PackageOptional bit NOT NULL DEFAULT 0, VersionOptional
bit NOT NULL DEFAULT 0
```

where "AppVManagement" is the name of the database.

Users cannot open a package in a user-published connection group if you add or remove an optional package

In environments that are running the RDS Client or that have multiple concurrent users per computer, logged-in users cannot open applications in packages that are in a user-published connection group if an optional package is added to or removed from the connection group.

Workaround: Have users log out and then log back in.

Error message is erroneously displayed when the connection group is published only to the user

When you run Repair-AppvClientConnectionGroup, the following error is displayed, even when the connection group is published only to the user: "Internal App-V Integration error: Package not integrated for the user. Please ensure that the package is added to the machine and published to the user."

Workaround: Do one of the following:

- Publish all packages in a connection group.

  The problem arises when the connection group being repaired has packages that are missing or not

available to the user (that is, not published globally or to the user). However, the repair will work if all of the connection group's packages are available, so ensure that all packages are published.

- Repair packages individually using the Repair-AppvClientPackage command rather than the Repair-AppvClientConnectionGroup command.

  Determine which packages are available to users and then run the **Repair-AppvClientPackage** command once for each package. Use Windows PowerShell cmdlets to do the following:

  1. Get all the packages in a connection group.
  2. Check to see if each package is currently published.
  3. If the package is currently published, run **Repair-AppvClientPackage** on that package.

Icons not displayed properly in Sequencer

Icons in the Shortcuts and File Type Associations tab are not displayed correctly when modifying a package in the App-V Sequencer. This problem occurs when the size of the icons are not 16x16 or 32x32.

Workaround: Only use icons that are 16x16 or 32x32.

InsertVersionInfo.sql script no longer required for the Management Database

The InsertVersionInfo.sql script is not required for versions of the App-V management database later than App-V 5.0 SP3.

The Permissions.sql script should be updated according to **Step 2** in KB article 3031340.

> **IMPORTANT**
>
> **Step 1** of the KB article listed above isn't required for versions of App-V later than App-V 5.0 SP3.

Microsoft Visual Studio 2012 not supported

App-V doesn't support Visual Studio 2012.

Workaround: Use a newer version of Microsoft Visual Studio.

Application filename restrictions for App-V Sequencer

The App-V Sequencer cannot sequence applications with filenames matching "CO_<x>" where x is any numeral. Error 0x8007139F will be generated.

Workaround: Use a different filename

Related resources list

For information that can help with troubleshooting App-V for Windows 10, see:

- Application Virtualization (App-V): List of Microsoft Support Knowledge Base Articles
- The Official Microsoft App-V Team Blog
- Technical Reference for App-V
- App-V TechNet Forum

For App-V issues, use the App-V TechNet Forum.

HELP US TO IMPROVE

Release Notes for App-V for Windows 10, version 1703

6/6/2019 • 2 minutes to read • Edit Online

Applies to

- Windows 10, version 1703

The following are known issues and workarounds for Application Virtualization (App-V) running on Windows 10, version 1703.

PROBLEM	WORKAROUND
Unable to manually create a system-owned folder needed for the `set-AppVClientConfiguration` PowerShell cmdlet when using the *PackageInstallationRoot*, *IntegrationRootUser*, or *IntegrationRootGlobal* parameters.	Don't create this file manually, instead let the `Add-AppVClientPackage` cmdlet auto-generate it.
Failure to update an App-V package from App-V 5.x to the latest in-box version, by using the PowerShell sequencing commands.	Make sure you have the complete App-V package or the MSI file from the original app.
Unable to modify the locale for auto-sequencing.	Open the `C:\Program Files (x86)\Windows Kits\10\Microsoft Application Virtualization\AutoSequencer\Unattend_Sequencer_User_Setup_Template` file and include the language code for your locale. For example, if you wanted Spanish (Spain), you'd use: **es-ES**.
Filetype and protocol handlers aren't registering properly with the Google Chrome browser, causing you to not see App-V packages as an option for default apps from the **Settings** > **Apps**> **Default Apps** area.	The recommended workaround is to add the following code to the AppXManifest.xml file, underneath the **<appv:Extensions>** tag:

PROBLEM	WORKAROUND

```
<appv:Extension Category="AppV.URLProtocol">
    <appv:URLProtocol>
        <appv:Name>ftp</appv:Name>
        <appv:ApplicationURLProtocol>
            <appv:DefaultIcon>
[{ProgramFilesX86}]\Google\Chrome\Application\chrome.
exe,0</appv:DefaultIcon>
            <appv:ShellCommands>

<appv:DefaultCommand>open</appv:DefaultCommand>
                <appv:ShellCommand>
                    <appv:ApplicationId>
[{ProgramFilesX86}]\Google\Chrome\Application\chrome.
exe</appv:ApplicationId>
                    <appv:Name>open</appv:Name>
                    <appv:CommandLine>"
[{ProgramFilesX86}]\Google\Chrome\Application\chrome.
exe" -- "%1"</appv:CommandLine>
                    <appv:DdeExec>
                        <appv:DdeCommand />
                    </appv:DdeExec>
                </appv:ShellCommand>
            </appv:ShellCommands>
        </appv:ApplicationURLProtocol>
    </appv:URLProtocol>
</appv:Extension>
<appv:Extension Category="AppV.URLProtocol">
    <appv:URLProtocol>
        <appv:Name>http</appv:Name>
        <appv:ApplicationURLProtocol>
            <appv:DefaultIcon>
[{ProgramFilesX86}]\Google\Chrome\Application\chrome.
exe,0</appv:DefaultIcon>
            <appv:ShellCommands>

<appv:DefaultCommand>open</appv:DefaultCommand>
                <appv:ShellCommand>
                    <appv:ApplicationId>
[{ProgramFilesX86}]\Google\Chrome\Application\chrome.
exe</appv:ApplicationId>
                    <appv:Name>open</appv:Name>
                    <appv:CommandLine>"
[{ProgramFilesX86}]\Google\Chrome\Application\chrome.
exe" -- "%1"</appv:CommandLine>
                    <appv:DdeExec>
                        <appv:DdeCommand />
                    </appv:DdeExec>
                </appv:ShellCommand>
            </appv:ShellCommands>
        </appv:ApplicationURLProtocol>
    </appv:URLProtocol>
</appv:Extension>
<appv:Extension Category="AppV.URLProtocol">
    <appv:URLProtocol>
        <appv:Name>https</appv:Name>
        <appv:ApplicationURLProtocol>
            <appv:DefaultIcon>
[{ProgramFilesX86}]\Google\Chrome\Application\chrome.
exe,0</appv:DefaultIcon>
            <appv:ShellCommands>

<appv:DefaultCommand>open</appv:DefaultCommand>
                <appv:ShellCommand>
                    <appv:ApplicationId>
[{ProgramFilesX86}]\Google\Chrome\Application\chrome.
exe</appv:ApplicationId>
                    <appv:Name>open</appv:Name>
                    <appv:CommandLine>"
[{ProgramFilesX86}]\Google\Chrome\Application\chrome.
exe" -- "%1"</appv:CommandLine>
                    <appv:DdeExec>
                        <appv:DdeCommand />
                    </appv:DdeExec>
                </appv:ShellCommand>
            </appv:ShellCommands>
        </appv:ApplicationURLProtocol>
    </appv:URLProtocol>
</appv:Extension>
```

Related resources list

For information that can help with troubleshooting App-V for Windows 10, see:

- Application Virtualization (App-V): List of Microsoft Support Knowledge Base Articles
- The Official Microsoft App-V Team Blog
- Technical Reference for App-V
- App-V TechNet Forum

For App-V issues, use the App-V TechNet Forum.

Related topics

- What's new in App-V for Windows 10
- Release Notes for App-V for Windows 10, version 1607

Evaluating App-V

5/31/2019 • 2 minutes to read • Edit Online

Applies to

- Windows 10, version 1607

Before you deploy App-V into a production environment, you should evaluate it in a lab environment. You can use the information in this topic to set up App-V in a lab environment for evaluation purposes only.

Configure lab computers for App-V Evaluation

Use the following links for information about setting up the App-V sequencer on a computer in your lab environment.

Installing the App-V Sequencer and Creating Packages

Use the following links for information about setting up the App-V sequencer and creating packages in your lab environment.

- How to Install the Sequencer
- Creating and Managing App-V Virtualized Applications

Configuring the App-V Server

Use the following links for information about setting up the App-V server in your lab environment.

- How to Deploy the App-V server
- Administering App-V Virtual Applications by Using the Management Console

Enabling the App-V Client

Use the following links for more information about creating and managing virtualized packages in your lab environment.

- Enable the App-V desktop client
- How to Configure the Client to Receive Package and Connection Groups Updates From the Publishing Server

Related topics

- Getting Started with App-V

High-level architecture for App-V

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Use the following information to simplify your Microsoft Application Virtualization (App-V) deployment.

Architecture overview

A typical App-V implementation consists of the following elements.

ELEMENT	DESCRIPTION
App-V Management server	The App-V Management server provides overall management functionality for the App-V infrastructure. Additionally, you can install more than one instance of the management server in your environment which provides the following benefits: **Fault tolerance and high availability**—installing and configuring the App-V Management server on two separate computers can help in situations when one of the servers is unavailable or offline. You can also help increase App-V availability by installing the Management server on multiple computers. In this scenario, consider using a network load balancer to keep server requests balanced. **Scalability**—you can add additional management servers as necessary to support a high load. For example, you can install multiple servers behind a load balancer.
App-V Publishing Server	The App-V publishing server provides functionality for virtual application hosting and streaming. The publishing server does not require a database connection and supports HTTP and HTTPS protocols. You can also help increase App-V availability by installing the Publishing server on multiple computers. You should also consider having a network load balancer to keep server requests balanced.
App-V Reporting Server	The App-V Reporting server lets authorized users run and view existing App-V reports and ad hoc reports for managing App-V infrastructure. The Reporting server requires a connection to the App-V reporting database. You can also help increase App-V availability by installing the Reporting server on multiple computers. You should also consider having a network load balancer to keep server requests balanced.
App-V Client	The App-V client enables packages created using App-V to run on target computers.

NOTE

If you are using App-V with electronic software distribution (ESD), you aren't required to use the App-V Management server. However, you can still use App-V's reporting and streaming functionality.

Related topics

- Getting Started with App-V

Planning for App-V

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Use the following information to plan to deploy App-V without disrupting your existing network or user experience.

Planning information

Preparing your environment for App-V describes the computing environment requirements and installation prerequisites that should be planned for before beginning App-V setup.

Planning to deploy App-V describes the minimum hardware and software requirements and other planning information for the App-V sequencer and App-V server components.

App-V planning checklist is a planning checklist that can assist you with App-V deployment planning.

Other App-V planning resources

- Application Virtualization (App-V) overview
- Getting started with App-V
- Deploying App-V
- Operations for App-V
- Troubleshooting App-V
- Technical reference for App-V

Preparing your environment for App-V

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

There are several different deployment configurations and prerequisites that you must consider before creating your deployment plan for Microsoft App-V. The following articles will help you gather the information you need to set up a deployment plan that best suits your business' needs.

App-V prerequisites

App-V prerequisites lists the prerequisite software that you must install before installing App-V.

App-V security considerations

App-V security considerations describes accounts, groups, log files, and other considerations for securing your App-V environment.

Other App-V planning resources

- Planning for App-V

App-V for Windows 10 prerequisites

5/31/2019 • 6 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Before installing App-V for Windows 10, ensure that you have installed all of the following required prerequisite software.

For a list of supported operating systems and hardware requirements for the App-V server, sequencer, and client, see App-V Supported Configurations.

Summary of software preinstalled on each operating system

The following table indicates the software that is already installed for different operating systems.

OPERATING SYSTEM	PREREQUISITE DESCRIPTION
Windows 10	All prerequisite software is already installed.
Windows 8.1	All prerequisite software is already installed. If you're running Windows 8, upgrade to Windows 8.1 before using App-V.
Windows Server 2016	The following prerequisite software is already installed: - Microsoft .NET Framework 4.5 - Windows PowerShell 3.0 Installing Windows PowerShell requires a restart.
Windows 7	No prerequisite software is installed. You must install the software before you can install App-V.

App-V Server prerequisite software

Install the required prerequisite software for the App-V server components.

What to know before you start

The account you use to install the App-V Server components must have:

- Administrative rights on the computer on which you are installing the components.
- The ability to query Active Directory Domain Services.

You must specify a port where each component will be hosted, and add the associated firewall rules to allow incoming requests to the specified ports.

Web Distributed Authoring and Versioning (WebDAV) is automatically disabled for the Management Service.

The following are supported deployment scenarios for App-V:

- A stand-alone deployment where all components are deployed on the same server.
- A distributed deployment.

The following deployment scenarios are not supported:

- Installing side-by-side instances of multiple App-V Server versions on the same server.
- Installing the App-V server components on a computer that runs server core or domain controller.

ITEM	DESCRIPTION
Account for installing the App-V Server	The account that you use to install the App-V Server components must have: - Administrative rights on the computer on which you are installing the components. - The ability to query Active Directory Domain Services.
Port and firewall	- Specify a port where each component will be hosted. - Add the associated firewall rules to allow incoming requests to the specified ports.
Web Distributed Authoring and Versioning (WebDAV)	WebDAV is automatically disabled for the Management Service.
Supported deployment scenarios	- A stand-alone deployment, where all components are deployed on the same server. - A distributed deployment.
Unsupported deployment scenarios	- Installing side-by-side instances of multiple App-V Server versions on the same server. - Installing the App-V server components on a computer that runs server core or domain controller.

Management server prerequisite software

PREREQUISITES AND REQUIRED SETTINGS	DETAILS
Supported version of SQL Server	For supported versions, see App-V supported configurations.
Microsoft .NET Framework 4.5.1 (Web Installer)	
Windows PowerShell 3.0	Installing Windows PowerShell 3.0 requires a restart.
Download and install KB2533623	Applies to Windows 7 only.
Visual C++ Redistributable Packages for Visual Studio 2013	
64-bit ASP.NET registration	
Windows Server Web Server Role	This role must be added to a server operating system that is supported for the Management server.
Web Server (IIS) Management Tools	Select **IIS Management Scripts and Tools**.

PREREQUISITES AND REQUIRED SETTINGS	DETAILS
Web Server Role Services	Common HTTP features: - Static content - Default document Application development: - ASP.NET - .NET Extensibility - ISAPI Extensions - ISAPI Filters Security: - Windows Authentication - Request Filtering Management Tools: - IIS Management Console
Default installation location	%PROGRAMFILES%\Microsoft Application Virtualization Server
Location of the Management database	SQL Server database name, SQL Server database instance name, and database name.
Management console and Management database permissions	A user or group that can access the Management console and database after the deployment is complete. Only these users or groups can access the Management console and database unless the Management console is used to add additional administrators.
Management service website name	Name for the Management console website.
Management service port binding	Unique port number for the Management service. This port cannot be used by another process on the computer.

> **IMPORTANT**
>
> JavaScript must be enabled on the browser that opens the Web Management Console.

Management server database prerequisite software

The Management database is only required if you use the App-V Management server.

PREREQUISITES AND REQUIRED SETTINGS	DETAILS
Microsoft .NET Framework 4.5.1 (Web Installer)	
Visual C++ Redistributable Packages for Visual Studio 2013	
Default installation location	%PROGRAMFILES%\Microsoft Application Virtualization Server

PREREQUISITES AND REQUIRED SETTINGS	DETAILS
Custom SQL Server instance name (if applicable)	Format to use: **INSTANCENAME** This format assumes that the installation is on the local computer. If you specify the name with the format **SVR\INSTANCE**, the installation will fail.
Custom database name (if applicable)	Unique database name. Default: AppVManagement
Management server location	Machine account on which the Management server is deployed. Format to use: **Domain\MachineAccount**
Management server installation administrator	Account used to install the Management server. Format to use: **Domain\AdministratorLoginName**
Microsoft SQL Server Service Agent	Configure the Management database computer so that the Microsoft SQL Server Agent service is restarted automatically. For instructions, see Configure SQL Server Agent to restart services automatically.

Publishing server prerequisite software

PREREQUISITES AND REQUIRED SETTINGS	DETAILS
Microsoft .NET Framework 4.5.1 (Web Installer)	
Visual C++ Redistributable Packages for Visual Studio 2013	
64-bit ASP.NET registration	
Web Server role	This role must be added to a server operating system that is supported for the Management server.
Web Server (IIS) Management Tools	Select **IIS Management Scripts and Tools**.
Web Server Role Services	Common HTTP features: - Static content - Default document Application development: - ASP.NET - .NET Extensibility - ISAPI Extensions - ISAPI Filters Security: - Windows Authentication - Request Filtering Management Tools: - IIS Management Console
Default installation location	%PROGRAMFILES%\Microsoft Application Virtualization Server

PREREQUISITES AND REQUIRED SETTINGS	DETAILS
Management service URL	URL of the App-V Management service. This is the port with which the Publishing server communicates. Management server and Publishing server are installed on the same server, use the format **https://localhost:12345**. If the Management server and Publishing server are installed on different servers, use the format **https://MyAppvServer.MyDomain.com**.
Publishing service website name	Name for the Publishing website.
Publishing service port binding	Unique port number for the Publishing service. This port cannot be used by another process on the computer.

Reporting server prerequisite software

PREREQUISITES AND REQUIRED SETTINGS	DETAILS
Supported version of SQL Server	For supported versions, see App-V supported configurations.
Microsoft .NET Framework 4.5.1 (Web Installer)	
Visual C++ Redistributable Packages for Visual Studio 2013	
64-bit ASP.NET registration	
Windows Server Web Server role	This role must be added to a server operating system that is supported for the Management server.
Web Server (IIS) Management Tools	Select **IIS Management Scripts and Tools**.
Web Server Role Services	To reduce the risk of unwanted or malicious data being sent to the Reporting server, you should restrict access to the Reporting Web Service per your corporate security policy. Common HTTP features: - Static content - Default document Application development: - ASP.NET - .NET Extensibility - ISAPI Extensions - ISAPI Filters Security: - Windows Authentication - Request Filtering - Management Tools: - IIS Management Console
Default installation location	%PROGRAMFILES%\Microsoft Application Virtualization Server
Reporting service website name	Name for the Reporting website.

PREREQUISITES AND REQUIRED SETTINGS	DETAILS
Reporting service port binding	Unique port number for the Reporting service. This port cannot be used by another process on the computer.

Reporting database prerequisite software

You only require the Reporting database if you're using the App-V Reporting server.

PREREQUISITES AND REQUIRED SETTINGS	DETAILS
Microsoft .NET Framework 4.5.1 (Web Installer)	
Visual C++ Redistributable Packages for Visual Studio 2013	
Default installation location	%PROGRAMFILES%\Microsoft Application Virtualization Server
Custom SQL Server instance name (if applicable)	Format to use: **INSTANCENAME** his format assumes that the installation is on the local computer. If you specify the name with the format **SVR\INSTANCE**, the installation will fail.
Custom database name (if applicable)	Unique database name. Default: AppVReporting
Reporting server location	The Reporting server will be deployed on this machine account. Format to use: **Domain\MachineAccount**
Reporting server installation administrator	Account used to install the Reporting server. Format to use: **Domain\AdministratorLoginName**
Microsoft SQL Server Service and Microsoft SQL Server Service Agent	Configure these services to be associated with user accounts that have access to query AD DS.

Sequencer prerequisite software

What to know before installing the prerequisites:

- Best practice: The computer that runs the sequencer should have the same hardware and software configurations as the computers running the virtual applications.
- The sequencing process is resource-intensive, so make sure that the computer running the sequencer has plenty of memory, a fast processor, and a fast hard drive. The system requirements of locally installed applications must not exceed those of the sequencer. For more information, see App-V supported configurations.

PREREQUISITE	DETAILS
Microsoft .NET Framework 4.5.1 (Web Installer)	
Windows PowerShell 3.0	Installing Windows PowerShell 3.0 requires a restart.

PREREQUISITE	DETAILS
KB2533623	Applies to Windows 7 only: download and install the KB.

Related topics

- Planning for App-V
- App-V Supported Configurations

App-V security considerations

6/10/2019 • 3 minutes to read • Edit Online

Applies to: Windows 10, version 1607

This topic contains a brief overview of the accounts and groups, log files, and other security-related considerations for Microsoft Application Virtualization (App-V).

> **IMPORTANT**
> App-V is not a security product and does not provide any guarantees for a secure environment.

The PackageStoreAccessControl (PSAC) feature has been deprecated

Effective as of June, 2014, the PackageStoreAccessControl (PSAC) feature introduced in Microsoft Application Virtualization (App-V) 5.0 Service Pack 2 (SP2) has been deprecated in both single-user and multi-user environments.

General security considerations

Understand the security risks. The most serious risk to App-V is from unauthorized users hijacking an App-V client's functionality, giving the hacker the ability to reconfigure key data on App-V clients. By comparison, short-term loss of App-V functionality from a denial-of-service attack would not be as catastrophic.

Physically secure your computers. A security strategy that doesn't consider physical security is incomplete. Anyone with physical access to an App-V server could potentially attack the entire client base, so potential physical attacks or thefts should be prevented at all cost. App-V servers should be stored in a physically secure server room with controlled access. Lock the computer with the operating system or a secured screen saver to keep computers secure when the administrators are away.

Apply the most recent security updates to all computers. To stay informed about the latest updates for operating systems, Microsoft SQL Server, and App-V, see the Microsoft Security TechCenter. (THIS LINK NEEDS TO BE UPDATED)

Use strong passwords or pass phrases. Always use strong passwords with 15 or more characters for all App-V and App-V administrator accounts. Never use blank passwords. For more information about password concepts, see Password Policy and Strong Passwords. (THIS LINK NEEDS TO BE UPDATED)

Accounts and groups in App-V

A best practice for user account management is to create domain global groups and add user accounts to them. After that, add the domain global accounts to the necessary App-V local groups on the App-V servers.

> **NOTE**
> App-V client computer accounts that need to connect to the publishing server must be part of the publishing server's **Users** local group. By default, all computers in the domain are part of the **Authorized Users** group, which is part of the **Users** local group.

App-V server security

No groups are created automatically during App-V setup. You should create the following Active Directory Domain Services global groups to manage App-V server operations.

GROUP NAME	DETAILS	IMPORTANT NOTES
App-V Management Admin group	Used to manage the App-V management server. This group is created during the App-V Management Server installation.	The management console can't create a new group after installation is complete.
Database read/write for Management Service account	Provides read/write access to the management database. This account should be created during App-V management database installation.	
App-V Management Service install admin account	Provides public access to schema-version table in management database. This account should be created during App-V management database installation.	This is only required if the management database is being installed separately from the service.
App-V Reporting Service install admin account	Public access to schema-version table in reporting database. This account should be created during the App-V reporting database installation.	This is only required if reporting database is being installed separately from the service.

Consider the following additional information:

- Access to the package shares: If a share exists on the same computer as the management Server, the **Network** service requires read access to the share. In addition, each App-V client computer must have read access to the package share.

> **NOTE**
> In previous versions of App-V, package share was referred to as content share.

- Registering publishing servers with Management Server: A publishing server must be registered with the Management server. For example, it must be added to the database, so that the Publishing server machine accounts are able to call into the Management service API.

App-V package security

The following will help you plan how to ensure that virtualized packages are secure.

- If an application installer applies an access control list (ACL) to a file or directory, then that ACL is not persisted in the package. If the file or directory is modified by a user when the package is deployed, the modified file or directory will either inherit the ACL in the **%userprofile%** or inherit the ACL of the target computer's directory. The former occurs if the file or directory does not exist in a virtual file system location; the latter occurs if the file or directory exists in a virtual file system location, such as **%windir%**.

App-V log files

During App-V setup, setup log files are created in the **%temp%** folder of the installing user.

Related topics

Preparing Your Environment for App-V

Planning to Deploy App-V for Windows 10

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

There are several different deployment configurations and requirements to consider before you deploy App-V for Windows 10. Review this topic for information about what you'll need to make a deployment plan that best meets your needs.

App-V supported configurations

App-V supported configurations describes the minimum hardware and operating system requirements for each App-V components. For information about software that you must install before you install App-V, see App-V Prerequisites.

App-V capacity planning

App-V capacity planning describes the available options for scaling your App-V deployment.

Planning for high availability with App-V

Planning for high availability with App-V describes the available options for ensuring high availability of App-V databases and services.

Planning to Deploy App-V with an Electronic Software Distribution System

Planning to Deploy App-V with an Electronic Software Distribution System describes the options and requirements for deploying App-V with an electronic software distribution system.

Planning for App-V server deployment

Planning for the App-V server deployment describes the planning considerations for the App-V Server components and their functions.

Planning for the App-V Sequencer and Client deployment

Planning for the App-V Sequencer and Client deployment describes planning considerations you should make for deploying the App-V Client and the Sequencer software, which you use to create virtual applications and application packages.

Planning for migrating from a previous version of App-V

Migrating to App-V from a previous version describes the recommended path for migrating from previous versions of App-V without disrupting your existing server configurations, packages, and clients.

Planning for using App-V with Office

Planning for using App-V with Office describes the requirements for using App-V with Office and the supported

scenarios, including information about coexisting versions of Office.

Planning to use folder redirection with App-V

Planning to use folder redirection with App-V explains how folder redirection works with App-V.

Other Resources for App-V Planning

- Planning for App-V
- Performance Guidance for Application Virtualization

App-V Supported Configurations

5/31/2019 • 3 minutes to read • Edit Online

Applies to: Windows 10, version 1607; Windows Server 2016; Windows Server 2012 R2; Windows Server 2012; Windows Server 2008 R2

This topic specifies the requirements to install and run App-V in your Windows 10 environment. For information about prerequisite software such as the .NET Framework, see App-V prerequisites.

App-V Server system requirements

This section lists the operating system and hardware requirements for all App-V server components.

Unsupported App-V server scenarios

The App-V server does not support the following scenarios:

- Deployment to a computer that runs the Server Core installation option.
- Deployment to a computer that runs a previous version of the App-V server components. You can only install App-V side-by-side with the App-V 4.5 Lightweight Streaming Server (LWS) server. This scenario doesn't support side-by-side deployment of App-V and the Application Virtualization Management Service (HWS) 4.x.
- Deployment to a computer running Microsoft SQL Server Express edition.
- Deployment to a domain controller.
- Short paths. If you plan to use a short path, you must create a new volume.

Management server operating system requirements

You can install the App-V Management server on a server running Windows Server 2008 R2 with SP1 or later.

> **IMPORTANT**
>
> Deploying a Management server role to a computer with Remote Desktop Services enabled is not supported.

Management server hardware requirements

- A 64-bit (x64) processor that runs at 1.4 GHz or faster.
- 1 GB RAM (64-bit).
- 200 MB of available hard disk space, not including the content directory.

Management server database requirements

The following table lists the SQL Server versions that the App-V Management database installation supports.

SQL SERVER VERSION	SERVICE PACK	SYSTEM ARCHITECTURE
Microsoft SQL Server 2014		32-bit or 64-bit
Microsoft SQL Server 2012	SP2	32-bit or 64-bit
Microsoft SQL Server 2008 R2	SP3	32-bit or 64-bit

Publishing server operating system requirements

The App-V Publishing server can be installed on a server that runs Windows Server 2008 R2 with SP1 or later.

Publishing server hardware requirements

App-V adds no additional requirements beyond those of Windows Server.

- A 64-bit (x64) processor that runs at 1.4 GHz or faster.
- 2 GB RAM (64-bit).
- 200 MB of available hard disk space, not including the content directory.

Reporting server operating system requirements

You can install the App-V Reporting server on a server running Windows Server 2008 R2 with SP1 or later.

Reporting server hardware requirements

App-V adds no additional requirements beyond those of Windows Server.

- A 64-bit (x64) processor that runs at 1.4 GHz or faster.
- 2 GB RAM (64-bit).
- 200 MB of available hard disk space, not including the content directory.

Reporting server database requirements

The following table lists the SQL Server versions that are supported for the App-V Reporting database installation.

SQL SERVER VERSION	SERVICE PACK	SYSTEM ARCHITECTURE
Microsoft SQL Server 2014		32-bit or 64-bit
Microsoft SQL Server 2012	SP2	32-bit or 64-bit
Microsoft SQL Server 2008 R2	SP3	32-bit or 64-bit

App-V client and Remote Desktop Services client requirements

With Windows 10, version 1607 and later releases, the App-V client is included with Windows 10 Enterprise and Windows 10 Education. The App-V client is no longer part of the Microsoft Desktop Optimization Pack. Before you can use the App-V client, it must be enabled, as described in Enable the App-V desktop client.

Similarly, the App-V Remote Desktop Services (RDS) client is included with Windows Server 2016 Standard and Windows Server 2016 Datacenter.

Sequencer system requirements

The following table lists the operating systems that the App-V Sequencer installation supports.

OPERATING SYSTEM	SERVICE PACK	SYSTEM ARCHITECTURE
Microsoft Windows Server 2012 R2		64-bit
Microsoft Windows Server 2012		64-bit
Microsoft Windows Server 2008 R2	SP1	64-bit
Microsoft Windows 10		32-bit and 64-bit

OPERATING SYSTEM	SERVICE PACK	SYSTEM ARCHITECTURE
Microsoft Windows 8.1		32-bit and 64-bit
Microsoft Windows 8		32-bit and 64-bit
Microsoft Windows 7	SP1	32-bit and 64-bit

Sequencer hardware requirements

See the Windows or Windows Server documentation for the hardware requirements.

Supported versions of System Center Configuration Manager

The App-V client works with System Center Configuration Manager versions starting with Technical Preview for System Center Configuration Manager, version 1606.

Related topics

- Planning to deploy App-V
- App-V prerequisites

App-V Capacity Planning

6/6/2019 • 12 minutes to read • Edit Online

Applies to: Windows Server 2016

The following recommendations can be used as a baseline to help determine capacity planning information that is appropriate to your organization's App-V infrastructure.

> **IMPORTANT**
> Use the information in this section only as a general guide for planning your App-V deployment. Your system capacity requirements will depend on the specific details of your hardware and application environment. Additionally, the performance numbers displayed in this document are examples and your results may vary.

Determine the project scope

Before you design the App-V infrastructure, determining which applications will be available virtually, and also identify the target users and their locations. This information will determine what type of App-V infrastructure your project should implement. Your should base your decisions about your project's scope on your organization's specific needs.

TASK	MORE INFORMATION
Determine application scope	The App-V infrastructure can be set up in different ways depending on which applications you want to virtualize. This means your first task is to define which applications you want to virtualize.
Determine location scope	"Location scope" refers to the physical locations where you plan to run the virtualized applications (for example, enterprise-wide or a specific geographic location). It can also refer to the user population that will run the virtual applications (for example, a single department). You should obtain a network map that includes the connection paths, the available bandwidth for each location, the number of users using virtualized applications, and the WAN link speed.

Determine which App-V infrastructure is required

You can also manage your App-V environment using an electronic software distribution (ESD) solution such as Microsoft Systems Center Configuration Manager. For more information see How to deploy App-V packages using electronic software distribution.

- **Standalone model**—The standalone model allows virtual applications to be Windows Installer-enabled for distribution without streaming. App-V in Standalone mode only needs the sequencer and the client; no additional components are required. Applications are prepared for virtualization using a process called sequencing. For more information, see Planning for the App-V Sequencer and Client deployment. The standalone model is recommended for the following scenarios:
  - When there are disconnected remote users who can't connect to the App-V infrastructure.
  - When you're running a software management system, such as System Center 2012 Configuration

Manager.
  - When network bandwidth limitations inhibit electronic software distribution.
- **Full infrastructure model**—The full infrastructure model provides for software distribution, management, and reporting capabilities; it also includes the streaming of applications across the network. The App-V full infrastructure model consists of one or more App-V management servers that can be used to publish applications to all clients. Publishing places the virtual application icons and shortcuts on the target computer. It can also stream applications to local users. For more information about how to install the management server, see Planning for App-V Server deployment. The full infrastructure model is recommended for the following scenarios:
  - When you want to use the Management Server to publish the application to target computers.
  - For rapid provisioning of applications to target computers.
  - When you want to use App-V reporting.

> **IMPORTANT**
> The App-V full infrastructure model requires Microsoft SQL Server to store configuration data. For more information, see App-V supported configurations.

End-to-end server sizing guidance

The following section describes end-to-end App-V sizing and planning. For more specific information, refer to the subsequent sections.

> **NOTE**
> Round trip response time on the client is the time taken by the computer running the App-V client to receive a successful notification from the publishing server. Round trip response time on the publishing server is the time taken by the computer running the publishing server to receive a successful package metadata update from the management server.

- 20,000 clients can target a single publishing server to obtain the package refreshes in an acceptable round trip time (<3 seconds).
- A single management server can support up to 50 publishing servers for package metadata refreshes in an acceptable round trip time (<5 seconds).

App-V Management Server capacity planning recommendations

The App-V publishing servers require the management server for package refresh requests and package refresh responses. The management server then sends the information to the management database to retrieve information. For more information about App-V management server supported configurations, see App-V supported configurations.

> **NOTE**
> The default refresh time on the App-V publishing server is ten minutes.

When multiple simultaneous publishing servers contact a single management server for package metadata refreshes, the following three factors will influence the publishing server's round-trip response time:

1. The number of publishing servers making simultaneous requests.
2. The number of connection groups configured on the management server.
3. The number of access groups configured on the management server.

The following table describes each factor that impacts round-trip time in more detail.

> **NOTE**
>
> Round trip response time is the time taken by the computer running the App-V publishing server to receive a successful package metadata update from the management server.

FACTORS IMPACTING ROUND-TRIP RESPONSE TIME	DESCRIPTION
The number of publishing servers simultaneously requesting package metadata refreshes.	A single management server can respond to up to 320 publishing servers simultaneously requesting publishing metadata. For example, in a case with 30 publishing servers simultaneously requesting publishing metadata, the round-trip response time is about 40 seconds, while for less than 50 servers it's less than 5 seconds. From 50 to 320 publishing servers, response team increases linearly (approximately 2×).
The number of connection groups configured on the management server.	For up to 100 connection groups, there is no significant change in the round-trip response time on the publishing server. For 100–400 connection groups, there is a minor linear increase in the round-trip response time.
The number of access groups configured on the management server.	For up to 40 access groups, there is a linear (approximately 3×) increase in the round-trip response time on the publishing server.

The following table displays sample values for each of the previous factors. In each variation, 120 packages are refreshed from the App-V management server.

SCENARIO	VARIATION	NUMBER OF CONNECTION GROUPS	NUMBER OF ACCESS GROUPS	NUMBER OF PUBLISHING SERVERS	NETWORK CONNECTION TYPE	ROUND-TRIP RESPONSE TIME (SECONDS)	MANAGEMENT SERVER CPU UTILIZATION
Publishing servers contact management server for publishing metadata at same time	Number of publishing servers.	0 0 0 0 0 0	1 1 1 1 1 1	50 100 200 300 315 320	LAN	5 10 19 32 30 37	17 17 17 15 17 15
Publishing metadata contains connection groups	Number of connection groups	10 20 100 150 300 400	1 1 1 1 1 1	100 100 100 100 100 100	LAN	10 11 11 16 22 25	17 19 22 19 20 20
Publishing metadata contains access groups	Number of access groups	0 0 0 0	1 10 20 40	100 100 100 100	LAN	10 43 153 535	17 26 24 24

The CPU utilization of the computer running the management server is around 25% irrespective of the number of publishing servers targeting it. The Microsoft SQL Server database transactions/sec, batch requests/sec and user

connections are identical irrespective of the number of publishing servers. For example, transactions/sec is approximately 30, batch requests approximately 200, and user connects approximately six.

Using a geographically distributed deployment, where the management server and publishing servers utilize a slow link network between them, the round-trip response time on the publishing servers is within acceptable time limits (<5 seconds), even for 100 simultaneous requests on a single management server.

SCENARIO	VARIATION	NUMBER OF CONNECTION GROUPS	NUMBER OF ACCESS GROUPS	NUMBER OF PUBLISHING SERVERS	NETWORK CONNECTION TYPE	ROUND-TRIP RESPONSE TIME (SECONDS)	MANAGEMENT SERVER CPU UTILIZATION (IN %)
Network connection between the publishing server and management server	1.5 Mbps Slow link Network	0 0	1 1	50 100	1.5 Mbps Cable DSL	4 5	1 2
Network connection between the publishing server and management server	LAN/WiFi Network	0 0	1 1	100 200	WiFi	11 20	15 17

Whether the management server and publishing servers are connected over a slow link network, or a high speed network, the management server can handle approximately 15,000 package refresh requests in 30 minutes.

App-V Reporting Server capacity planning recommendations

App-V clients send reporting data to the reporting server. The reporting server then records the information in the Microsoft SQL Server database and returns a successful notification back to the computer running App-V client. For more information about the App-V Reporting Server's supported configurations see App-V supported configurations.

> **NOTE**
> Round-trip response time is the time taken by the computer running the App-V client to send the reporting information to the reporting server and receive a successful notification from the reporting server.

SCENARIO	SUMMARY
Multiple App-V clients send reporting information to the reporting server simultaneously.	Round-trip response time from the reporting server is 2.6 seconds for 500 clients. Round-trip response time from the reporting server is 5.65 seconds for 1000 clients. Round-trip response time increases linearly depending on number of clients.

SCENARIO	SUMMARY
Requests per second processed by the reporting server.	A single reporting server and a single database, can process a maximum of 139 requests per second. The average is 121 requests/second. Using two reporting servers reporting to the same Microsoft SQL Server database, the average requests/second, like a single reporting server, is about 127, with a max of 278 requests/second. A single reporting server can process 500 concurrent/active connections. A single reporting server can process a maximum 1,500 concurrent connections.
Reporting database.	Lock contention on the computer running Microsoft SQL Server is the limiting factor for requests/second. Throughput and response time are independent of database size.

Calculating random delay

The random delay specifies the maximum delay (in minutes) for data to be sent to the reporting server. When the scheduled task is started, the client generates a random delay between **0** and **ReportingRandomDelay** and will wait the specified duration before sending data.

Random delay = 4 × number of clients/average requests per second.

Example: Random delay for 500 clients with 120 requests per second is *4 × 500/120 = about 17 minutes*.

App-V publishing server capacity planning recommendations

Computers running the App-V client connect to the App-V publishing server to send a publishing refresh request and receive a response. Round trip response time is measured on the computer running the App-V client, while processor time is measured on the publishing server. For more information about App-V Publishing Server supported configurations, see App-V supported configurations.

> **IMPORTANT**
>
> The following list displays the main factors to consider when setting up the App-V publishing server:
>
> - The number of clients connecting simultaneously to a single publishing server.
> - The number of packages in each refresh.
> - The available network bandwidth in your environment between the client and the App-V publishing server.

SCENARIO	SUMMARY
Multiple App-V clients connect to a single publishing server simultaneously.	A publishing server running dual core processors can respond to at most 5000 clients requesting a refresh simultaneously. For 5,000–10,000 clients, the publishing server requires a minimum quad core. For 10,000–20,000 clients, the publishing server should have dual quad cores for more efficient response times. A publishing server with a quad core can refresh up to 10,000 packages within three seconds. (Supports 10,000 simultaneous clients.)
Number of packages in each refresh.	Increasing number of packages will increase response time by about 40% (up to 1,000 packages).

SCENARIO	SUMMARY
Network between the App-V client and the publishing server.	Across a slow network (1.5 Mbps bandwidth), there is a 97% increase in response time compared to LAN (up to 1,000 users).

> **NOTE**
> The publishing server CPU usage is always high during the time interval when it must process simultaneous requests (>90% in most cases). The publishing server can handle about 1,500 client requests in one second.

SCENARIO	VARIATION	NUMBER OF APP-V CLIENTS	NUMBER OF PACKAGES	PROCESSOR CONFIGURATION ON PUBLISHING SERVER	NETWORK CONNECTION TYPE	APP-V CLIENT ROUND-TRIP TIME (IN SECONDS)	PUBLISHING SERVER CPU UTILIZATION (IN %)
App-V client sends publishing refresh request and receives response, each request containing 120 packages	Number of clients	100 1,000 5,000 10,000	120 120 120 120	Dual Core Dual Core Quad Core Quad Core	LAN	1 2 2 3	100 99 89 77
Multiple packages in each refresh.	Number of packages	1,000 1,000	500 1,000	Quad Core	LAN	2 3	92 91
Network between client and publishing server.	1.5 Mbps Slow link network	100 500 1,000	120 120 120	Quad Core	1.5 Mbps intra-continental network	3 10 (0.2% failure rate) 7 (1% failure rate)	

App-V streaming capacity planning recommendations

Computers running the App-V client stream the virtual application package from the streaming server. Round trip response time is measured on the computer running the App-V client, and is the time taken to stream the entire package.

> **IMPORTANT**
> The following list identifies the main factors to consider when setting up the App-V streaming server:
>
> - The number of clients streaming application packages simultaneously from a single streaming server.
> - The size of the package being streamed.
> - The available network bandwidth in your environment between the client and the streaming server.

SCENARIO	SUMMARY
Multiple App-V clients stream applications from a single streaming server simultaneously.	If the number of clients simultaneously streaming from the same server increases, there is a linear relationship with the package download/streaming time.
Size of the package being streamed.	The package size has a significant impact on the streaming/download time only for larger packages with a size of about 1 GB. For package sizes ranging from 3 MB to 100 MB, the streaming time ranges from 20 seconds to 100 seconds, with 100 simultaneous clients.
Network between the App-V client and the streaming server.	Across a slow network (1.5 Mbps bandwidth), there is a 70–80% increase in response time compared to LAN (up to 100 users).

The following table displays sample values for each of the factors in the previous list:

SCENARIO	VARIATION	NUMBER OF APP-V CLIENTS	SIZE OF EACH PACKAGE	NETWORK CONNECTION TYPE	ROUND-TRIP TIME ON THE APP-V CLIENT (IN SECONDS)
Multiple App-V clients streaming virtual application packages from a streaming server.	Number of clients.	100 200 1,000 100 200 1,000	3.5 MB 3.5 MB 3.5 MB 5 MB 5 MB 5 MB	LAN	29 39 391 35 68 461
Size of each package being streamed.	Size of each package.	100 200 100 200	21 MB 21 MB 109 MB 109 MB	LAN	33 83 100 160
Network connection between client and App-V streaming server.	1.5 Mbps Slow link network.	100 100	3.5 MB 5 MB	1.5 Mbps intra-continental network	102 121

Each App-V streaming server should be able to handle a minimum of 200 clients concurrently streaming virtualized applications.

> **NOTE**
>
> The actual time to it will take to stream is determined primarily by the number of clients streaming simultaneously, number of packages, package size, the server's network activity, and network conditions.

For example, an average user can stream a 100 MB package in less than 2 minutes, when 100 simultaneous clients are streaming from the server. However, a package of size 1 GB could take up to 30 minutes. In most real-world environments, streaming demand is not uniformly distributed, you will need to understand the approximate peak streaming requirements present in your environment to properly size the number of required streaming servers.

The number of clients a streaming server can support can be significantly increased and the peak streaming requirements reduced if you pre-cache your applications. You can also increase the number of clients a streaming server can support by using on-demand streaming delivery and stream optimized packages.

Combining App-V server roles

Discounting scaling and fault-tolerance requirements, the minimum number of servers that a location with Active Directory connectivity needs to function is one. This server will host the management server, management server service, and Microsoft SQL Server roles. This means that you can arrange server roles in any combination you like, as they don't conflict with one another.

Ignoring scaling requirements, the minimum number of servers that a fault-tolerant implementation needs to function is four. The management server and Microsoft SQL Server roles support placement in fault-tolerant configurations. The management server service can be combined with any of the roles, but remains a single point of failure.

Although there are many fault-tolerance strategies and technologies you can use, not all are applicable to a given service. Additionally, if App-V roles are combined, the resulting incompatibilities could cause certain fault-tolerance options to stop working.

Related topics

- App-V supported configurations
- Planning for high availability with App-V
- Planning to deploy App-V

Planning for high availability with App-V Server

6/10/2019 • 3 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Microsoft Application Virtualization (App-V) system configurations can take advantage of options that maintain a high available service level.

The following sections will he following sections to help you understand the options to deploy App-V in a highly available configuration.

Support for Microsoft SQL Server clustering

You can run the App-V Management and Reporting databases on computers running Microsoft SQL Server clusters. However, you must install the databases using scripts.

For deployment instructions, see How to deploy the App-V databases by using SQL scripts.

Support for IIS network load balancing

You can use Internet Information Services' (IIS) network load balancing (NLB) to configure a highly available environment for computers running the App-V Management, Publishing, and Reporting services that are deployed through IIS.

Review the following articles to learn more about configuring IIS and NLB for computers running Windows Server operating systems:

- Achieving High Availability and Scalability - ARR and NLB describes how to configure IIS 7.0.
- Network load balancing overview will tell you more about how to configure Microsoft Windows Server.

  This information also applies to IIS NLB clusters in Windows Server 2008, Windows Server 2008 R2, or Windows Server 2012.

> **NOTE**
>
> The IIS NLB functionality in Windows Server 2012 is generally the same as in Windows Server 2008 R2. However, some task details have changed in Windows Server 2012. To learn how to work with these changes, see Common management tasks and navigation in Windows.

Support for clustered file servers when running SCS mode

Running App-V Server in Shared Content Store (SCS) mode with clustered file servers is supported.

To enable SCS mode configurations, follow these steps:

1. Configure the App-V client to run in SCS mode. For more information, see Deploying the App-V Sequencer and Configuring the Client.
2. Configure the file server cluster, configured in either the scale out mode (which started with Windows Server 2012) or the earlier clustering mode, with a virtual SAN.

The following steps can be used to validate the configuration:

1. Add a package on the publishing server. To learn how to add a package, see How to add or upgrade packages by using the Management console.
2. Perform a publishing refresh on the computer running the App-V client and open an application.
3. Switch cluster nodes mid-publishing refresh and mid-streaming to ensure failover works correctly.

Review the following articles to learn more about configuring Windows Server failover clusters:

- Create a failover cluster
- Use cluster shared volumes in a failover cluster

Support for Microsoft SQL Server mirroring

Using Microsoft SQL Server mirroring, where the App-V management server database is mirrored utilizing two SQL Server instances, for App-V management server databases is supported.

Review the following to learn more about how to configure Microsoft SQL Server mirroring:

- Prepare a mirror database for mirroring (SQL Server)
- Establish a database mirroring session using Windows Authentication (SQL Server Management Studio) (FIX LINK)

The following steps can be used to validate the configuration:

1. Initiate a Microsoft SQL Server Mirroring session.
2. Select **Failover** to designate a new master Microsoft SQL Server instance.
3. Verify that the App-V management server continues to function as expected after the failover.

The connection string on the management server can be modified to include `failover partner = <server2>`. This will only help when the primary on the mirror has failed over to the secondary and the computer running the App-V client is doing a fresh connection (say after reboot).

Use the following steps to modify the connection string to include `failover partner = <server2>`:

> **IMPORTANT**
> This process involves changing the Windows registry with Registry Editor. If you change the Windows registry incorrectly, you can cause serious problems that might require you to reinstall Windows. Always make a backup copy of the registry files (**System.dat** and **User.dat**) before changing the registry. Microsoft can't guarantee that problems caused by changing the registry can be resolved, so change the registry at your own risk.

1. Log in to the management server and open **regedit**.
2. Navigate to **HKEY_LOCAL_MACHINE** \ **Software** \ **Microsoft** \ **AppV** \ **Server** \ **ManagementService**.
3. Modify the **MANAGEMENT_SQL_CONNECTION_STRING** value with the `failover partner = <server2>` value.
4. Restart management service using the IIS console.

> **NOTE**
> Database Mirroring is on the list of deprecated database engine features in SQL Server 2012 due to the **AlwaysOn** feature available starting with Microsoft SQL Server 2012.

Click any of the following links for more information:

- Prepare a mirror database for mirroring (SQL Server).

- Establish a database mirroring session using Windows Authentication (SQL Server Management Studio).
- Deprecated database engine features in SQL Server 2012.

Support for Microsoft SQL Server Always On configuration

The App-V management server database supports deployments to computers running Microsoft SQL Server with the **Always On** configuration. For more information, see Always On Availability Groups (SQL Server).

Related topics

- Planning to deploy App-V

Planning to Deploy App-V with an electronic software distribution system

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

If you are using an electronic software distribution (ESD) system to deploy App-V packages, review the following planning considerations. For information about deploying App-V with System Center Configuration Manager, see Introduction to application management in Configuration Manager.

Review the following component and architecture requirements options that apply when you use an ESD to deploy App-V packages:

DEPLOYMENT REQUIREMENT OR OPTION	DESCRIPTION
The App-V Management server, Management database, and Publishing server are not required.	These functions are handled by the implemented ESD solution.
You can deploy the App-V Reporting server and Reporting database side-by-side with the ESD.	The side-by-side deployment lets you collect data and generate reports. If you enable the App-V client to send report information without using the App-V Reporting server, the reporting data will be stored in associated .xml files.

Related topics

- Planning to deploy App-V
- How to deploy App-V packages Using Electronic Software Distribution
- How to enable only administrators to publish packages by using an ESD

Planning for the App-V server deployment

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows Server 2016

The Microsoft Application Virtualization (App-V) server infrastructure consists of a set of specialized features that can be installed on one or more server computers, based on the requirements of the enterprise.

About the App-V server

The App-V server consists of the following features:

- Management Server—provides overall management functionality for the App-V infrastructure.
- Management Database—facilitates database predeployments for App-V management.
- Publishing Server—provides hosting and streaming functionality for virtual applications.
- Reporting Server—provides App-V reporting services.
- Reporting Database—facilitates database predeployments for App-V reporting.

The following list describes recommended App-V server infrastructure installation methods:

- Install the App-V server. For more information, see How to deploy the App-V Server.
- Install the database, reporting, and management features on separate computers. For more information, see How to install the Management and Reporting databases on separate computers from the Management and Reporting services.
- Use Electronic Software Distribution (ESD). For more information, see How to deploy App-V packages using Electronic Software Distribution.
- Install all server features on a single computer.

App-V server interaction

This section describes how the various App-V server roles interact with each other.

The App-V Management Server contains the repository of packages and their assigned configurations. For Publishing Servers that are registered with the Management Server, the associated metadata is provided to the Publishing servers for use when publishing refresh requests are received from computers running the App-V Client. App-V publishing servers managed by a single management server can serve different clients with different website names and port bindings. Additionally, all Publishing Servers managed by the same Management Server are replicas of each other.

> **NOTE**
>
> The Management Server does not perform load balancing. The associated metadata is passed to the publishing server for use when processing client requests.

Server-related protocols and external features

The following table lists server-related protocols used by the App-V servers, and also describes the reporting mechanism for each server type.

SERVER TYPE	PROTOCOLS	EXTERNAL FEATURES NEEDED	REPORTING
IIS server	HTTP HTTPS	This server-protocol combination requires a mechanism to synchronize content between the Management Server and the Streaming Server. When using HTTP or HTTPS, use an IIS server and a firewall to protect the server from exposure to the Internet.	Internal
File	SMB	This server-protocol combination requires support to synchronize the content between the Management Server and the Streaming Server. Use a client computer that's capable of file sharing or streaming.	Internal

Related topics

- Planning to deploy App-V
- Deploying the App-V server

Planning for the App-V Sequencer and Client Deployment

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Before you can use App-V, you must install the App-V Sequencer and enable the App-V client. You can also the App-V shared content store, although it isn't required. The following sections will tell you how to set these up.

Planning for App-V Sequencer deployment

App-V uses a process called sequencing to create virtualized applications and application packages. Sequencing requires the use of a computer that runs the App-V Sequencer.

> **NOTE**
> For information about the new functionality of App-V sequencer, see What's new in App-V.

The computer running the App-V sequencer must meet the minimum system requirements. For a list of these requirements, see App-V supported configurations.

Ideally, you should install the sequencer on a computer running as a virtual machine. This lets you revert the computer that's running the sequencer to a “clean” state before sequencing another application. When installing the sequencer using a virtual machine, you should do the following things:

1. Install all associated sequencer prerequisites.
2. Install the sequencer.
3. Take a “snapshot” of the environment.

> **IMPORTANT**
> Your corporate security team should review and approve the sequencing process plan before implementing it. For security reasons, it's a good idea to keep sequencer operations in a lab separate from the production environment. The sequencing computers must be capable of connecting to the corporate network to copy finished packages to the production servers. However, because the sequencing computers are typically operated without antivirus protection, they shouldn't remain on the corporate network unprotected. You can protect your sequencing computers by operating them on an isolated network, behind a firewall, or by using virtual machines on an isolated virtual network. Make sure your solution follows your company's corporate security policies.

Planning for App-V client deployment

In Windows 10, version 1607, the App-V client is included with the operating system. For more information, see Enable the App-V desktop client.

Planning for the App-V Shared Content Store (SCS)

The App-V Shared Content Store mode allows computers running the App-V client to run virtualized applications without saving any package contents to the App-V client computer. Virtual applications are streamed to target

computers only when requested by the client.

The following list displays some of the benefits of using App-V SCS:

- Reduced app-to-app and multi-user application conflicts reduces the need for regression testing
- Reduced deployment risk accelerates application deployment
- Simplified profile management

Other App-V deployment resources

- Planning to deploy App-V

Related topics

- How to install the sequencer
- Enable the App-V desktop client
- Deploying the App-V Sequencer and configuring the client

Planning for deploying App-V with Office

6/6/2019 • 6 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Use the following information to plan how to deploy Office within Microsoft Application Virtualization (App-V).

App-V language pack support

You can use the App-V Sequencer to create plug-in packages for language packs, language interface packs, proofing tools, and ScreenTip languages. You can then include the plug-in packages in a connection group, along with the Office package that you create by using the Office Deployment Toolkit. The Office applications and the plug-in language packs will interact seamlessly in the same connection group.

> **NOTE**
>
> Microsoft Visio and Microsoft Project do not support the Thai Language Pack.

Supported versions of Microsoft Office

For a list of supported Office products, see Microsoft Office Product IDs that App-V supports.

> **NOTE**
>
> You must use the Office Deployment Tool instead of the App-V Sequencer to create App-V packages for Office 365 ProPlus. App-V does not support package creation for volume-licensed versions of Office Professional Plus or Office Standard. Support for the Office 2013 version of Office 365 ended in February 2017.

Using App-V with coexisting versions of Office

You can simultaneously install more than one version of Microsoft Office on the same computer with a feature called "Microsoft Office coexistence." You can implement Office coexistence with combinations of all major versions of Office and with installation methods, as applicable, by using the Windows Installer-based (MSI) version of Office, Click-to-Run, and App-V. However, Microsoft doesn't recommend using Office coexistence.

Microsoft's recommended best practice is to avoid Office coexistence completely to prevent compatibility issues. However, in cases where issues arise during migration that you can't immediately resolve, Office coexistence can allow for faster migration to the latest Office version. Since this solution is only meant to be temporary, your organization must set up a plan to fully transition to the newer version of Office in the meantime.

Before you implement Office coexistence

Before implementing Office coexistence, review the information in the following table that corresponds to the newest version of Office that you will use in coexistence. The documentation linked here will guide you in implementing coexistence for Windows Installer-based (MSI) and Click-to-Run installations of Office.

OFFICE VERSION	RELEVANT HOW-TO GUIDES

OFFICE VERSION	RELEVANT HOW-TO GUIDES
Office 2016	How to use Outlook 2016 or 2013 and an earlier version of Outlook installed on the same computer
Office 2013	How to use Office 2013 suites and programs (MSI deployment) on a computer running another version of Office
Office 2010	How to use Office 2010 suites and programs on a computer running another version of Office] (https://support.microsoft.com/kb/2121447)

Once you've reviewed the relevant guide, this topic will supplement what you've learned with information about Office coexistence that's more specific to App-V deployments.

Supported Office coexistence scenarios

The following tables summarize supported coexistence scenarios. They are organized according to the version and deployment method you're starting with and the version and deployment method you are migrating to. Be sure to fully test all coexistence solutions before deploying them to a production audience.

> **NOTE**
> Microsoft does not support the use of multiple versions of Office in Windows Server environments that have the Remote Desktop Session Host role service enabled. To run Office coexistence scenarios, you must disable this role service.

Windows integrations and Office coexistence

Windows Installer-based and Click-to-Run Office installation methods integrate with certain points of the underlying Windows OS, but coexistence can cause these integrations to conflict. App-V can sequence certain version of Office to exclude integrations that could be potential problem spots, isolating them from the OS and preventing compatibility or user experience issues.

The following table describes the integration level of each version of Office, and which mode App-V can use to sequence them.

OFFICE VERSION	THE MODES APP-V CAN SEQUENCE THIS VERSION OF OFFICE WITH
Office 2007	Always non-integrated. App-V does not offer any operating system integrations with a virtualized version of Office 2007.
Office 2010	Integrated and non-integrated mode.
Office 2013	Always integrated. Windows operating system integrations cannot be disabled.
Office 2016	Always integrated. Windows operating system integrations cannot be disabled.

Microsoft recommends deploying Office coexistence with only one integrated Office instance. For example, if you're using App-V to deploy Office 2010 and Office 2013, you should sequence Office 2010 in non-integrated mode. For more information about sequencing Office in non-integration (isolated) mode, see How to sequence Microsoft Office 2010 in Microsoft Application Virtualization 5.0.

Known limitations of Office coexistence scenarios

The following sections describe issues you might encounter when using App-V to implement coexistence with

Office.

Limitations common to Windows Installer-based/Click-to-Run and App-V Office coexistence scenarios

Limitations can occur when you install the following versions of Office on the same computer:

- Office 2010 with the Windows Installer-based version
- Office 2013 or Office 2016 with App-V

Publishing Office 2013 or Office 2016 with App-V at the same time as an earlier version of the Windows Installer-based Office 2010 might cause the Windows Installer to start. This is because either the Windows Installer-based or Click-to-Run version of Office 2010 is trying to automatically register itself to the computer.

To bypass the auto-registration operation for native Word 2010, follow these steps:

1. Exit Word 2010.
2. Start the Registry Editor by doing the following:
   - In Windows 7k, select **Start**, type **regedit** in the Start Search box, then select the Enter key.
   - In Windows 8.1 or Windows 10, enter **regedit**, select **Enter** on the Start page, then select the Enter key.

     If you're prompted for an administrator password, enter the password. If you're prompted for a confirmation, select **Continue**.
3. Locate and then select the following registry subkey:

   ```
   HKEY_CURRENT_USER\Software\Microsoft\Office\14.0\Word\Options
   ```
4. On the **Edit** menu, select **New**, then select **DWORD Value**.
5. Type **NoReReg**, then select the Enter key.
6. Right-click **NoReReg**, then select **Modify**.
7. In the **Valuedata** box, enter **1**, then select **OK**.
8. On the File menu, select **Exit** to close Registry Editor.

How Office integrates with Windows when you use App-V to deploy Office

When you deploy Office 2013 or Office 2016 with App-V, Office is fully integrated with the operating system that provides end-users with the same features and functionality that Office has when deployed without App-V.

The Office 2013 or Office 2016 App-V package supports the following integration points with the Windows operating system:

INTEGRATION POINT	DESCRIPTION
Skype for Business (formerly Lync) Meeting Join plug-in for Firefox and Chrome	User can join Skype meetings from Firefox and Chrome
Sent to OneNote Print Driver	User can print to OneNote
OneNote Linked Notes	OneNote Linked Notes

INTEGRATION POINT	DESCRIPTION
Send to OneNote Internet Explorer add-in	User can send to OneNote from IE
Firewall exception for Skype for Business (formerly Lync) and Outlook	Firewall exception for Skype for Business (formerly Lync) and Outlook
MAPI client	Native apps and add-ins can interact with virtual Outlook through MAPI
SharePoint Plug-in for Firefox	User can use SharePoint features in Firefox
Mail Control Panel Applet	User gets the mail control panel applet in Outlook
Primary Interop Assemblies	Support managed add-ins
Office Document Cache Handler	Allows Document Cache for Office applications
Outlook Protocol Search Handler	User can search in Outlook
Active X Controls	For more information on ActiveX controls, refer to ActiveX Control API Reference.
OneDrive Pro Icon Overlays	Windows Explorer shell icon overlays when users look at folders OneDrive Pro folders
Shell extensions	
Shortcuts	
Windows Search	

Related topics

- Deploying Microsoft Office 2016 by Using App-V
- Deploying Microsoft Office 2013 by Using App-V
- Deploying Microsoft Office 2010 by Using App-V

Planning to Use Folder Redirection with App-V

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Microsoft Application Virtualization (App-V) supports the use of folder redirection, a feature that enables users and administrators to redirect the path of a folder to a new location.

What is folder redirection?

Folder redirection lets end users work with files that have been redirected to another folder as if the files still exist on the local drive.

- Users and administrators can redirect the path of a folder to a network location. The documents in the specified folder will be available to users from any computer in the network.
  - For example, you can redirect the Documents folder from your computer's local hard disk to a network location. The user can then access the folder's documents from any computer on the network.
- The new location can be a folder on either the local computer or a shared network.
- Folder redirection immediately updates the files, while roaming data is typically synchronized when the user logs in or out of a session.

Requirements for using folder redirection with App-V

To use %AppData% folder redirection, you must:

- Have an App-V package that has an AppData virtual file system (VFS) folder.
- Enable folder redirection and redirect users' folders to a shared folder, typically a network folder.
- Roam both or neither of the following:
  - Files under %appdata%\Microsoft\AppV\Client\Catalog
  - Registry settings under HKEY_CURRENT_USER\Software\Microsoft\AppV\Client\Packages

For more information, see Application publishing and client interaction.

Unsupported scenarios for App-V folder redirection

The following scenarios aren't supported by App-V:

- Configuring %LocalAppData% as a network drive.
- Redirecting the Start menu to a single folder for multiple users.
- If roaming AppData (%AppData%) is redirected to a network share that is not available, App-V applications will fail to launch, unless the unavailable network share has been enabled for Offline Files.

How to configure folder redirection for use with App-V

Folder redirection can be applied to different folders, such as Desktop, My Documents, My Pictures, and so on. However, the only folder that impacts the use of App-V applications is the user's roaming AppData folder (%AppData%). You can apply folder redirection to any other supported folders without impacting App-V.

How folder redirection works with App-V

The following table describes how folder redirection works when %AppData% is redirected to a network and when you have met the requirements listed earlier in this article.

VIRTUAL ENVIRONMENT STATE	ACTION THAT OCCURS
When the virtual environment starts.	The virtual file system (VFS) AppData folder is mapped to the local AppData folder (%LocalAppData%) instead of to the user's roaming AppData folder (%AppData%). - LocalAppData contains a local cache of the user's roaming AppData folder for the package in use. The local cache is located under `%LocalAppData%\Microsoft\AppV\Client\VFS\PackageGUID\AppData` - The latest data from the user's roaming AppData folder is copied to and replaces the data currently in the local cache. - While the virtual environment is running, data continues to be saved to the local cache. Data is served only out of %LocalAppData% and is not moved or synchronized with %AppData% until the end user shuts down the computer. - Entries to the AppData folder are made using the user context, not the system context.
When the virtual environment shuts down.	The local cached data in AppData (roaming) is zipped up and copied to the "real" roaming AppData folder in %AppData%. A time stamp that indicates the last known upload is simultaneously saved as a registry key under `HKCU\Software\Microsoft\AppV\Client\Packages\<PACKAGE_GUID>\AppDataTime`. App-V keeps the three most recent copies of the compressed data under %AppData% for redundancy.

App-V Planning Checklist

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

This checklist can be used to help you plan for preparing your organization for an App-V deployment.

> **NOTE**
>
> This checklist outlines the recommended steps and a high-level list of items to consider when planning an App-V deployment. It's a good idea to copy this checklist and customize it for your use.

STATUS	TASK	REFERENCES	NOTES
☐	Review the getting started information about App-V to gain a basic understanding of the product before beginning deployment planning.	Getting started with App-V	
☐	Plan for App-V deployment prerequisites and prepare your computing environment.	App-V prerequisites	
☐	If you plan to use the App-V management server, plan for the required roles.	Planning for the App-V server deployment	
☐	Plan for the App-V sequencer and client to create and run virtualized applications.	Planning for the App-V Sequencer and client deployment	
☐	If applicable, review the options and steps for migrating from a previous version of App-V.	Migrating to App-V from a previous version	
☐	Decide whether to configure App-V clients in Shared Content Store mode.	Deploying the App-V Sequencer and configuring the client	

Related topics

Planning for App-V

Deploying App-V for Windows 10

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

App-V supports several different deployment options. Review this topic for information about the tasks that you must complete at different stages in your deployment.

App-V Deployment Information

- Deploying the App-V Sequencer and configuring the client

  This section describes how to install the App-V sequencer used to virtualize applications, and how to enable the App-V client that runs on target computers to facilitate virtualized packages.

- Deploying the App-V server

  This section provides information about installing the App-V management, publishing, database, and reporting severs.

- App-V deployment checklist

  This section provides a deployment checklist that can be used to assist with installing App-V.

The following sections describe how to use App-V to deliver Microsoft Office as a virtualized application to computers in your organization.

- Deploying Microsoft Office 2016 by using App-V
- Deploying Microsoft Office 2013 by using App-V
- Deploying Microsoft Office 2010 by using App-V

Other App-V deployment resources

- Application Virtualization (App-V) overview
- Getting started with App-V
- Planning for App-V
- Operations for App-V
- Troubleshooting App-V
- Technical reference for App-V

Deploying the App-V Sequencer and configuring the client

5/31/2019 • 3 minutes to read • Edit Online

Applies to: Windows 10, version 1607

The App-V Sequencer and client let administrators to virtualize and run virtual applications.

Enable the client

The App-V client is the component that runs a virtualized application on a target computer. The client lets users interact with icons and file types, starting virtualized applications. The client can also get the virtual application content from the management server.

> **NOTE**
> In Windows 10, version 1607, App-V is included with the operating system. You only need to enable it.

Enable the App-V desktop client

Client configuration settings

The App-V client stores its configuration in the registry. Understanding the format used in the data registry can help you learn useful information about the client. For information about client settings that you can configure through Windows PowerShell or through the registry, see About client configuration settings.

Configure the client by using the ADMX template and Group Policy

You can use Group Policy to configure the client settings for the App-V client and the Remote Desktop Services client.

To manage the ADMX template, perform the following steps on the computer that you will use to manage Group Policy. This is typically the Domain Controller.

1. Save the **.admx** file to the following directory: `Windows\PolicyDefinitions`
2. Save the **.adml** file to the following directory: `Windows\PolicyDefinitions\<Language Directory>`

After you have completed the preceding steps, you can use Group Policy to configure the client settings by using the Group Policy Management Console under **Computer Configuration** > **Administrative Templates** > **System** > **App-V**.

Understanding Shared Content Store mode for App-V clients

App-V Shared Content Store (SCS) mode lets SCS App-V clients run virtualized applications without having to save any of the associated package data locally. All required virtualized package data is transmitted across the network; therefore, you should only use SCS mode in environments with a fast connection. Both the Remote Desktop Services (RDS) and the standard version of the App-V client are supported with SCS mode.

> **IMPORTANT**
> If the App-V client is configured to run in the SCS mode, the location where the App-V packages are streamed from must be available, otherwise, the virtualized package will fail. Additionally, we do not recommend deployment of virtualized applications to computers that run the App-V client in the SCS mode across the internet.

Additionally, the SCS is not a physical location that contains virtualized packages. It is a mode that allows the App-V client to stream the required virtualized package data across the network.

The SCS mode is helpful in the following scenarios:

- Virtual desktop infrastructure (VDI) deployments
- Remote Desktop Services deployments

To use SCS in your environment, you must configure the App-V client to run in SCS mode, as it does not use SCS mode by default.

There might be cases when the administrator pre-loads some virtual applications on the computer that runs the App-V client in SCS mode. This can be accomplished with Windows PowerShell commands to add, publish, and mount the package. For example, if a package is pre-loaded on all computers, the administrator could add, publish, and mount the package by using Windows PowerShell commands. The package would not stream across the network because it would be locally stored.

Configure the Group Policy setting for the SCS Mode for App-V clients

Use the following steps to locate and configure the Group Policy setting for the SCS Mode for App-V clients.

1. In the Group Policy Management Console, navigate to **Computer Configuration** > **Administrative Templates** > **System** > **App-V** > **Streaming**.
2. Enable the **Set the Shared Content Mode (SCS) mode** setting.

Configure an individual client to use SCS mode

To configure the App-V client to run in SCS mode, on the client, enter the following Windows PowerShell command:

```
Set-AppvClientConfiguration -SharedContentStoreMode 1
```

Deploy the Sequencer

The Sequencer is a tool that is used to convert standard applications into virtual packages for deployment to computers that run the App-V client. The Sequencer helps provide a simple and predictable conversion process with minimal changes to prior sequencing workflows. In addition, the Sequencer allows users to more easily configure applications to enable connections of virtualized applications.

For a list of changes in the App-V Sequencer, see What's new in App-V.

To deploy the sequencer, see How to install the Sequencer.

App-V client and Sequencer logs

You can use the App-V Sequencer log information to troubleshoot Sequencer installation and operational events while using App-V. The Sequencer-related log information can be reviewed with the **Event Viewer**. The following file path is the specific path for Sequencer-related events:

Event Viewer\Applications and Services Logs\Microsoft\App V.

NOTE

Sequencer-related events are prepended with **AppV_Sequencer**. Client-related events are prepended with **AppV_Client**.

About Client Configuration Settings

6/26/2019 • 8 minutes to read • Edit Online

Applies to: Windows 10, version 1607

The Microsoft Application Virtualization (App-V) client stores its configuration in the registry. Understanding how the register's format for data works can help you better understand the client, as you can configure many client actions by changing registry entries. This topic lists the App-V client configuration settings and explains their uses. You can use Windows PowerShell to modify the client configuration settings. For more information about using Windows PowerShell and App-V see Administering App-V by using Windows PowerShell.

You can use Group Policy to configure App-V client settings by navigating to the **Group Policy management console** at **Computer Configuration** > **Administrative Templates** > **System** > **App-V**.

App-V Client Configuration Settings: Windows PowerShell

The following table provides information about App-V client configuration settings that can be configured through Windows PowerShell cmdlets:

WINDOWS POWERSHELL CMDLET OR CMDLETS, OPTION TYPE	DESCRIPTION	DISABLED POLICY STATE KEYS AND VALUES
Set-AppvClientConfiguration, Set-AppvPublishingServer **-PackageInstallationRoot** String	Specifies directory where all new applications and updates will be installed.	Policy value not written (same as Not Configured)
Set-AppvClientConfiguration, Set-AppvPublishingServer **-PackageSourceRoot** String	Overrides source location for downloading package content.	Policy value not written (same as Not Configured)
Set-AppvClientConfiguration, Set-AppvPublishingServer **-AllowHighCostLaunch** True (enabled); False (Disabled state)	This setting controls whether virtualized applications are launched on Windows 10 machines connected by a metered network connection (for example, 4G).	0
Set-AppvClientConfiguration, Set-AppvPublishingServer **-ReestablishmentRetries** Integer (0–99)	Specifies the number of times to retry a dropped session.	Policy value not written (same as Not Configured)
Set-AppvClientConfiguration, Set-AppvPublishingServer **-ReestablishmentInterval** Integer (0–3600)	Specifies the number of seconds between attempts to reestablish a dropped session.	Policy value not written (same as Not Configured)

WINDOWS POWERSHELL CMDLET OR CMDLETS, OPTION TYPE	DESCRIPTION	DISABLED POLICY STATE KEYS AND VALUES
Set-AppvClientConfiguration, Set-AppvPublishingServer **-LocationProvider** String	Specifies the CLSID for a compatible implementation of the IAppvPackageLocationProvider interface.	Policy value not written (same as Not Configured)
Set-AppvClientConfiguration, Set-AppvPublishingServer **-CertFilterForClientSsl** String	Specifies the path to a valid certificate in the certificate store.	Policy value not written (same as Not Configured)
Set-AppvClientConfiguration, Set-AppvPublishingServer **-VerifyCertificateRevocationList** True (enabled); False (Disabled state)	Verifies Server certificate revocation status before streaming with HTTPS.	0
Set-AppvClientConfiguration, Set-AppvPublishingServer **-SharedContentStoreMode** True (enabled); False (Disabled state)	Specifies that streamed package contents will be not be saved to the local hard disk.	0
Set-AppvPublishingServer **-Name** String	Displays the name of publishing server.	Policy value not written (same as Not Configured)
Set-AppvPublishingServer **-URL** String	Displays the URL of publishing server.	Policy value not written (same as Not Configured)
Set-AppvPublishingServer **-GlobalRefreshEnabled** True (enabled); False (Disabled state)	Enables global publishing refresh (Boolean)	False
Set-AppvPublishingServer **-GlobalRefreshOnLogon** True (enabled); False (Disabled state)	Triggers a global publishing refresh on sign in. (Boolean)	False
Set-AppvPublishingServer **-GlobalRefreshInterval** Integer (0–744)	Specifies the publishing refresh interval using the GlobalRefreshIntervalUnit. To disable package refresh, specify 0.	0
Set-AppvPublishingServer **-GlobalRefreshIntervalUnit** 0 for hour, 1 for day	Specifies the interval unit (Hour 0–23, Day 0–31).	1

WINDOWS POWERSHELL CMDLET OR CMDLETS, OPTION TYPE	DESCRIPTION	DISABLED POLICY STATE KEYS AND VALUES
Set-AppvPublishingServer **-UserRefreshEnabled** True (enabled); False (Disabled state)	Enables user publishing refresh (Boolean)	False
Set-AppvPublishingServer **-UserRefreshOnLogon** True (enabled); False (Disabled state)	Triggers a user publishing refresh on sign in. (Boolean) Word count (with spaces): 60	False
Set-AppvPublishingServer **-UserRefreshInterval** Word count (with spaces): 85 Integer (0–744 Hours)	Specifies the publishing refresh interval using the UserRefreshIntervalUnit. To disable package refresh, select 0.	0
Set-AppvPublishingServer **-UserRefreshIntervalUnit** 0 for hour, 1 for day	Specifies the interval unit (Hour 0–23, Day 0–31).	1
Set-AppvClientConfiguration, Set-AppvPublishingServer **-MigrationMode** True (enabled state); False (Disabled state)	Migration mode allows the App-V client to modify shortcuts and FTA's for packages created by a previous version of App-V.	
Set-AppvClientConfiguration, Set-AppvPublishingServer **-EnablePackageScripts** True (enabled); False (Disabled state)	Enables scripts defined in the package manifest of configuration files that should run.	
Set-AppvClientConfiguration **-RoamingFileExclusions** String	Specifies the file paths relative to %userprofile% that do not roam with a user's profile. For example, `/ROAMINGFILEEXCLUSIONS='desktop;my pictures'`	
Set-AppvClientConfiguration, Set-AppvPublishingServer **-RoamingRegistryExclusions** String	Specifies the registry paths that do not roam with a user profile. For example, `/ROAMINGREGISTRYEXCLUSIONS=software\\classes;software\\clients`	Policy value not written (same as Not Configured)

WINDOWS POWERSHELL CMDLET OR CMDLETS, OPTION TYPE	DESCRIPTION	DISABLED POLICY STATE KEYS AND VALUES
Set-AppvClientConfiguration, Set-AppvPublishingServer **-IntegrationRootUser** String	Specifies the location to create symbolic links associated with the current version of a per-user published package. All virtual application extensions, such as shortcuts and file type associations, will point to this path. If you don't specify a path, symbolic links will not be used when you publish the package. For example, `%localappdata%\\Microsoft\\AppV\\Client\\Integration` .	Policy value not written (same as Not Configured)
Set-AppvClientConfiguration, Set-AppvPublishingServer **-IntegrationRootGlobal** String	Specifies the location to create symbolic links associated with the current version of a globally published package. All virtual application extensions, such as shortcuts and file type associations, will point to this path. If you don't specify a path, symbolic links will not be used when you publish the package. For example, `%allusersprofile%\\Microsoft\\AppV\\Client\\Integration` .	Policy value not written (same as Not Configured)
Set-AppvClientConfiguration, Set-AppvPublishingServer **-VirtualizableExtensions** String	A comma-delineated list of file name extensions that can be used to determine if a locally installed application can be run in the virtual environment. When shortcuts, FTAs, and other extension points are created during publishing, App-V will compare the file name extension to the list if the application associated with the extension point is locally installed. If the extension is located, the **RunVirtual** command-line parameter will be added, and the application will run virtually. For more information about the **RunVirtual** parameter, see Running a locally installed application inside a virtual environment with virtualized applications.	Policy value not written
Set-AppvClientConfiguration, Set-AppvPublishingServer **-ReportingEnabled** True (enabled); False (Disabled state)	Returns information to a reporting server.	False
Set-AppvClientConfiguration, Set-AppvPublishingServer **-ReportingServerURL** String	Specifies the location on the reporting server where client information is saved.	Policy value not written (same as Not Configured)

WINDOWS POWERSHELL CMDLET OR CMDLETS, OPTION TYPE	DESCRIPTION	DISABLED POLICY STATE KEYS AND VALUES
Set-AppvClientConfiguration, Set-AppvPublishingServer **-ReportingDataCacheLimit** Integer [0–1024]	Specifies the maximum size in megabytes (MB) of the XML cache for storing reporting information. The size applies to the cache in memory. When the limit is reached, the log file will roll over. Set between 0 and 1024.	Policy value not written (same as Not Configured)
Set-AppvClientConfiguration, Set-AppvPublishingServer **-ReportingDataBlockSize** Integer [1024 - Unlimited]	Specifies the maximum size in bytes to transmit to the server for reporting upload requests. This can help avoid permanent transmission failures when the log has reached a significant size. Set between 1024 and unlimited.	Policy value not written (same as Not Configured)
Set-AppvClientConfiguration, Set-AppvPublishingServer **-ReportingStartTime** Integer (0–23)	Specifies the time to initiate the client to send data to the reporting server. You must specify a valid integer between 0–23 corresponding to the hour of the day. By default the **ReportingStartTime** will start on the current day at 10 P.M.or 22. **Note** You should configure this setting to a time when computers running the App-V client are least likely to be offline.	Policy value not written (same as Not Configured)
Set-AppvClientConfiguration, Set-AppvPublishingServer **-ReportingInterval** Integer	Specifies the retry interval that the client will use to resend data to the reporting server.	Policy value not written (same as Not Configured)
Set-AppvClientConfiguration, Set-AppvPublishingServer **-ReportingRandomDelay** Integer [0 - ReportingRandomDelay]	Specifies the maximum delay (in minutes) for data to be sent to the reporting server. When the scheduled task is started, the client generates a random delay between 0 and **ReportingRandomDelay** and will wait the specified duration before sending data. This can help to prevent collisions on the server.	Policy value not written (same as Not Configured)
Set-AppvClientConfiguration, Set-AppvPublishingServer **-EnableDynamicVirtualization** 1 (Enabled), 0 (Disabled)	Enables supported Shell Extensions, Browser Helper Objects, and Active X controls to be virtualized and run with virtual applications.	
Set-AppvClientConfiguration, Set-AppvPublishingServer **-EnablePublishingRefreshUI** 1 (Enabled), 0 (Disabled)	Enables the publishing refresh progress bar for the computer running the App-V Client.	

WINDOWS POWERSHELL CMDLET OR CMDLETS, OPTION TYPE	DESCRIPTION	DISABLED POLICY STATE KEYS AND VALUES
Sync-AppvPublishingServer **-HidePublishingRefreshUI** 1 (Enabled), 0 (Disabled)	Hides the publishing refresh progress bar.	
Set-AppvClientConfiguration, Set-AppvPublishingServer **-ProcessesUsingVirtualComponents** String	Specifies a list of process paths (that may contain wildcards) that are candidates for using dynamic virtualization (such as supported shell extensions, browser helper objects, and ActiveX controls). Only processes whose full path matches one of these items can use dynamic virtualization.	Empty string.

App-V client configuration settings: registry keys

The following table provides information about App-V client configuration settings that can be configured through the registry:

SETTING NAME TYPE	REGISTRY KEY VALUE	DISABLED POLICY STATE KEYS AND VALUES
PackageInstallationRoot String	Streaming\PackageInstallationRoot	Policy value not written (same as Not Configured)
PackageSourceRoot String	Streaming\PackageSourceRoot	Policy value not written (same as Not Configured)
AllowHighCostLaunch True (Enabled); False (Disabled state)	Streaming\AllowHighCostLaunch	0
ReestablishmentRetries Integer (0–99)	Streaming\ReestablishmentRetries	Policy value not written (same as Not Configured)
ReestablishmentInterval Integer (0–3600)	Streaming\ReestablishmentInterval	Policy value not written (same as Not Configured)
LocationProvider String	Streaming\LocationProvider	Policy value not written (same as Not Configured)
CertFilterForClientSsl String	Streaming\CertFilterForClientSsl	Policy value not written (same as Not Configured)
VerifyCertificateRevocationList True (Enabled); False (Disabled state)	Streaming\VerifyCertificateRevocationList	0
SharedContentStoreMode True (Enabled); False (Disabled state)	Streaming\SharedContentStoreMode	0
Name String	Publishing\Servers{serverId}\FriendlyName	Policy value not written (same as Not Configured)

SETTING NAME TYPE	REGISTRY KEY VALUE	DISABLED POLICY STATE KEYS AND VALUES
URL String	Publishing\Servers{serverId}\URL	Policy value not written (same as Not Configured)
GlobalRefreshEnabled True (Enabled); False (Disabled state)	Publishing\Servers{serverId}\GlobalEnabled	False
GlobalRefreshOnLogon True (Enabled); False (Disabled state)	Publishing\Servers{serverId}\GlobalLogonRefresh	False
GlobalRefreshInterval Integer (0–744)	Publishing\Servers{serverId}\GlobalPeriodicRefreshInterval	0
GlobalRefreshIntervalUnit 0 for hour, 1 for day	Publishing\Servers{serverId}\GlobalPeriodicRefreshIntervalUnit	1
UserRefreshEnabled True (Enabled); False (Disabled state)	Publishing\Servers{serverId}\UserEnabled	False
UserRefreshOnLogon True (Enabled); False (Disabled state)	Publishing\Servers{serverId}\UserLogonRefresh	False
UserRefreshInterval Word count (with spaces): 85; Integer (0–744 Hours)	Publishing\Servers{serverId}\UserPeriodicRefreshInterval	0
UserRefreshIntervalUnit 0 for hour, 1 for day	Publishing\Servers{serverId}\UserPeriodicRefreshIntervalUnit	1
MigrationMode True(Enabled state); False (Disabled state)	Coexistence\MigrationMode	
EnablePackageScripts True (Enabled); False (Disabled state)	\Scripting\EnablePackageScripts	
RoamingFileExclusions String		
RoamingRegistryExclusions String	Integration\RoamingRegistryExclusions	Policy value not written (same as Not Configured)
IntegrationRootUser String	Integration\IntegrationRootUser	Policy value not written (same as Not Configured)
IntegrationRootGlobal String	Integration\IntegrationRootGlobal	Policy value not written (same as Not Configured)
VirtualizableExtensions String	Integration\VirtualizableExtensions	Policy value not written
ReportingEnabled True (Enabled); False (Disabled state)	Reporting\EnableReporting	False

SETTING NAME TYPE	REGISTRY KEY VALUE	DISABLED POLICY STATE KEYS AND VALUES
ReportingServerURL String	Reporting\ReportingServer	Policy value not written (same as Not Configured)
ReportingDataCacheLimit Integer [0–1024]	Reporting\DataCacheLimit	Policy value not written (same as Not Configured)
ReportingDataBlockSize Integer [1024–Unlimited]	Reporting\DataBlockSize	Policy value not written (same as Not Configured)
ReportingStartTime Integer (0–23)	Reporting\ StartTime	Policy value not written (same as Not Configured)
ReportingInterval Integer	Reporting\RetryInterval	Policy value not written (same as Not Configured)
ReportingRandomDelay Integer [0 - ReportingRandomDelay]	Reporting\RandomDelay	Policy value not written (same as Not Configured)
EnableDynamicVirtualization 1 (Enabled), 0 (Disabled)	HKEY_LOCAL_MACHINE\Software\Microsoft\AppV\Client\Virtualization	
EnablePublishingRefreshUI 1 (Enabled), 0 (Disabled)	HKEY_LOCAL_MACHINE\Software\Microsoft\AppV\Client\Publishing	
HidePublishingRefreshUI 1 (Enabled), 0 (Disabled)		
ProcessesUsingVirtualComponents String	Virtualization\ProcessesUsingVirtualComponents	Empty string.

Related topics

- Deploying the App-V Sequencer and Configuring the Client

Enable the App-V in-box client

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

The App-V client is the component that runs virtualized applications on user devices. Once you enable the client, users can interact with icons and file names to start virtualized applications. The client can also get virtual application content from the management server.

With Windows 10, version 1607, the App-V client is installed automatically. However, you'll still need to enable the client yourself to allow user devices to access and run virtual applications. You can set up the client with the Group Policy editor or with Windows PowerShell.

Here's how to enable the App-V client with Group Policy:

1. Open the device's **Group Policy Editor**.
2. Navigate to **Computer Configuration** > **Administrative Templates** > **System** > **App-V**.
3. Run **Enables App-V Client**, then select **Enabled**.
4. Restart the device.

Here's how to enable the App-V client with Windows PowerShell:

1. Open Windows PowerShell.
2. Enter **Enable-Appv**, then select the Enter key.
3. Restart the device.
4. To verify that the App-V client is working, enter **Get-AppvStatus**, then select the Enter key.

Check out these articles for more information about how to configure the App-V client:

- Deploying the App-V Sequencer and configuring the client
- How to modify client configuration by using Windows PowerShell
- Using the client management console
- How to configure the client to receive package and connection group updates from the Publishing server

Install the App-V Sequencer

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

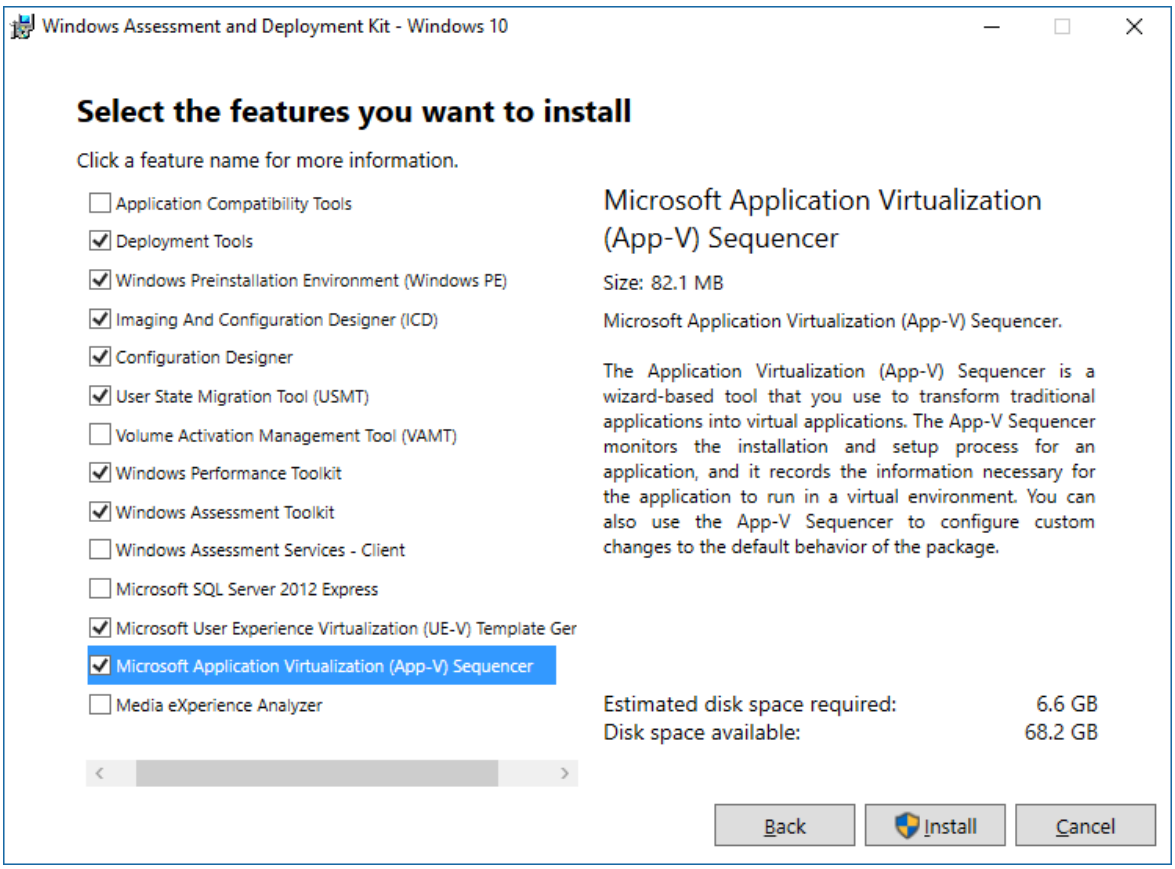
Use the App-V Sequencer to convert Win32 applications into virtual packages for deployment to user devices. Those devices must be running the App-V client to allow users to interact with virtual applications.

The App-V Sequencer is included in the Windows 10 Assessment and Deployment Kit (Windows ADK).

> **NOTE**
>
> The computer that will run the sequencer must not have the App-V client enabled. As a best practice, choose a computer with the same hardware and software configurations as the computers that will run the virtual applications. The sequencing process is resource-intensive, so make sure the computer that will run the Sequencer has plenty of memory, a fast processor, and a fast hard drive.

How to install the App-V Sequencer

1. Go to Download the Windows ADK.
2. Select the **Get Windows ADK for Windows 10** button on the page to start the ADK installer. Make sure that **Microsoft Application Virtualization (App-V) Sequencer** is selected during the installation.



3. To open the Sequencer, go to the **Start** menu and select **Microsoft Application Virtualization (App-V) Sequencer**.

See Creating and managing virtual applications and the Application Virtualization Sequencing Guide for information about creating virtual applications with the Sequencer.

Command-line options for installing the sequencer

You can also use the command line to install the App-V sequencer. The following list displays information about options for installing the sequencer using the command line and **appv_sequencer_setup.exe**:

COMMAND	DESCRIPTION
/INSTALLDIR	Specifies the installation directory.
/Log	Specifies where the installation log will be saved. The default location is **%Temp%**. For example, **C:\Logs\log.log**.
/q	Specifies a quiet or silent installation.
/Uninstall	Specifies the removal of the sequencer.
/ACCEPTEULA	Accepts the license agreement. This is required for an unattended installation. For example, **/ACCEPTEULA** or **/ACCEPTEULA=1**.
/LAYOUT	Specifies the associated layout action. It also extracts the Windows Installer (.msi) and script files to a folder without installing App-V. No value is expected.
/LAYOUTDIR	Specifies the layout directory. Requires a string value. For example, **/LAYOUTDIR="C:\Application Virtualization Client"**.
/? or **/h** or **/help**	Displays associated help.

To troubleshoot the App-V sequencer installation

For more information regarding the sequencer installation, you can view the error log in the **%temp%** folder. To review the log files, click **Start**, type **%temp%**, and then look for the **appv_ log**.

Related topics

- Planning to deploy App-V

Deploying the App-V server

6/10/2019 • 5 minutes to read • Edit Online

Applies to: Windows Server 2016

You can install the Application Virtualization (App-V) server components using different deployment configurations, which are described in this topic. Before you install the server features, review the server section of App-V security considerations.

> **NOTE**
> If you plan to use the App-V server components in your deployment, note that the version number is still listed as App-V 5.x, as the App-V server components have not changed in App-V for Windows 10.

To learn more about deploying App-V for Windows 10, read What's new in App-V.

> **IMPORTANT**
> Before installing and configuring the App-V servers, you must specify the port or ports where each component will be hosted. You must also add the associated firewall rules to allow incoming requests to access the specified ports, as the installer does not modify firewall settings.

Download and install App-V server components

> **NOTE**
> If you're already using App-V 5.x, you don't need to re-deploy the App-V server components, as they haven't changed since App-V 5.0 was released.

App-V offers the following five server components, each of which serves a specific purpose in an App-V environment.

- **Management server.** Use the App-V management server and console to manage your App-V infrastructure. See Administering App-V with the management console for more information about the management server.

> **NOTE**

```
>If you are using App-V with your electronic software distribution solution, you don’t need to use the
management server and console. However, you may want to take advantage of the reporting and streaming
capabilities in App-V.
```

- **Management database.** Use the App-V management database to facilitate database pre-deployments for App-V management. For more information about the management database, see How to deploy the App-V server.
- **Publishing server.** Use the App-V publishing server to host and stream virtual applications. The publishing server supports the HTTP and HTTPS protocols and does not require a database connection. To learn how to

configure the publishing server, see How to install the App-V publishing server.

- **Reporting server.** Use the App-V reporting server to generate reports that help you manage your App-V infrastructure. The reporting server requires a connection to the reporting database. To learn more about App-V's reporting capabilities, see About App-V reporting.
- **Reporting database.** Use the App-V reporting database to facilitate database pre-deployments for App-V reporting. To learn more about the reporting database, see How to deploy the App-V server.

All five App-V server components are included in the Microsoft Desktop Optimization Pack (MDOP) 2015 ISO package, which can be downloaded from either of the following locations:

- The MSDN (Microsoft Developer Network) subscriptions site. You must have a MSDN subscription to download the MDOP ISO package from this site.
- The Volume Licensing Service Center if you're using Windows 10 for Enterprise or Education.

In large organizations, you might want to install more than one instance of the server components to get the following benefits.

- Fault tolerance for situations when one of the servers is unavailable.
- High availability to balance server requests. A network load balancer can also help you achieve this.
- Scalability to support high loads. For example, you can install additional servers behind a network load balancer.

App-V standalone deployment

The App-V standalone deployment's topology is good for small deployments or test environments. In this implementation type, all server components are installed on a single computer. The services and associated databases will compete for the resources on the computer that runs the App-V components. However, because services and associated databases will compete for the computer's resources, it's not a good idea to use the standalone deployment for larger deployments.

The following articles will give you more information about how to set up an App-V standalone deployment.

- How to deploy the App-V server
- How to deploy the App-V server using a script

App-V server distributed deployment

The distributed deployment topology can support a large App-V client base, allowing you to more easily manage and scale your environment. When you use this type of deployment the App-V server components are deployed across multiple computers, based on your organization's structure and requirements.

- How to install the management and reporting databases on separate computers from the management and reporting services
- How to install the management server on a standalone computer and connect it to the database
- How to deploy the App-V server using a script
- How to install the publishing server on a remote computer
- How to install the management server on a standalone computer and connect it to the database

Using an Enterprise Software Distribution (ESD) solution and App-V

You can also deploy packages with an ESD. Its full integration capabilities will vary depending on which ESD you use.

> **NOTE**
> The App-V reporting server and reporting database can still be deployed alongside the ESD to collect the reporting data from the App-V clients. However, the other three server components should not be deployed, because they will conflict with the ESD functionality.

- Deploying App-V packages by Using Electronic Software Distribution (ESD)

App-V Server logs

You can use App-V server log information to help troubleshoot the server installation and operational events while using App-V. The server-related log information can be reviewed with the **Event Viewer**. The following line displays the specific path for Server-related events:

Event Viewer \ Applications and Services Logs \ Microsoft \ App V

Associated setup logs are saved in the following directory:

%temp%

App-V reporting

App-V reporting allows App-V clients to collect data and then send it back to be stored in a central repository. You can use this information to get a better view of the virtual application usage within your organization. The following list displays some of the types of information the App-V client collects:

- Information about the computer running the App-V client.
- Information about virtualized packages on a specific computer running the App-V client.
- Information about package open and shutdown for a specific user.

The reporting information will be maintained until it is successfully sent to the reporting server database. After the data is in the database, you can use Microsoft SQL Server Reporting Services (SSRS) to generate any necessary reports.

If you want to retrieve report information, you must use Microsoft SQL SSRS, which is available with Microsoft SQL. SSRS must be deployed separately to generate the associated reports, as it isn't automatically installed during App-V server installation.

For more information, see About App-V reporting and How to enable reporting on the App-V client by using Windows PowerShell.

Other App-V server resources

- Deploying App-V

How to Deploy the App-V Server (new installation)

6/10/2019 • 4 minutes to read • Edit Online

Applies to: Windows Server 2016

Before you start

> **IMPORTANT**
>
> If you're already using App-V 5.x, you don't need to re-deploy the App-V server components as they haven't changed since App-V 5.0 was released.

- Ensure that you've installed required software. See App-V prerequisites.
- Review the server section of App-V security considerations.
- Specify a port where each component will be hosted.
- Add firewall rules to allow incoming requests to access the specified ports.
- If you use SQL scripts instead of the Windows Installer to set up the Management database or Reporting database, you must run the required SQL scripts before installing the Management Server or Reporting Server. See How to deploy the App-V databases by using SQL scripts.

Installing the App-V server

1. Download the App-V server components. All five App-V server components are included in the Microsoft Desktop Optimization Pack (MDOP) 2015 ISO package, which can be downloaded from either of the following locations:
   - The MSDN (Microsoft Developer Network) subscriptions site. You must have a MSDN subscription to download the MDOP ISO package from this site.
   - The Volume Licensing Service Center if you're using Windows 10 for Enterprise or Education.
2. Copy the App-V server installation files to the computer on which you want to install it.
3. Start the App-V server installation by right-clicking and running **appv_server_setup.exe** as an administrator, and then click **Install**.
4. Review and accept the license terms, and choose whether to enable Microsoft updates.
5. On the **Feature Selection** page, select all components listed in the following table.

COMPONENT	DESCRIPTION
Management server	Provides overall management functionality for the App-V infrastructure.
Management database	Facilitates database predeployments for App-V management.
Publishing server	Provides hosting and streaming functionality for virtual applications.

COMPONENT	DESCRIPTION
Reporting server	Provides App-V reporting services.
Reporting database	Facilitates database predeployments for App-V reporting.

6. On the **Installation Location** page, accept the default location where the selected components will be installed, or change the location by typing a new path on the **Installation Location** line.

7. On the initial **Create New Management Database** page, configure the **Microsoft SQL Server instance** and **Management Server database** by selecting the appropriate option below.

METHOD	WHAT YOU NEED TO DO
You are using a custom Microsoft SQL Server instance.	Select **Use the custom instance**, then specify the instance name. Use the format **INSTANCENAME**. The assumed installation location is the local computer. Not supported: A server name using the format **ServerName\INSTANCE**.
You are using a custom database name.	Select **Custom configuration** and type the database name. The database name must be unique, or the installation will fail.

8. On the **Configure** page, accept the default value, **Use this local computer**.

> **NOTE**
>
> If you're installing the Management server and Management database side-by-side, the appropriate options are selected by default and cannot be changed.

9. On the initial **Create New Reporting Database** page, configure the **Microsoft SQL Server instance** and **Reporting Server database** by selecting the appropriate option below.

METHOD	WHAT YOU NEED TO DO
You are using a custom Microsoft SQL Server instance.	Select **Use the custom instance**, and type the name of the instance. Use the format **INSTANCENAME**. The assumed installation location is the local computer. Not supported: A server name using the format **ServerName\INSTANCE**.
You are using a custom database name.	Select **Custom configuration** and type the database name. The database name must be unique, or the installation will fail.

10. On the **Configure** page, accept the default value: **Use this local computer**.

> **NOTE**

```
>If you're installing the Management server and Management database side-by-side, the appropriate options are selected by default and cannot be changed.
```

11. On the **Configure** (Management Server Configuration) page, specify the following:

ITEM TO CONFIGURE	DESCRIPTION AND EXAMPLES
Specify AD group	Specify the AD group with sufficient permissions to manage the App-V environment. Example: MyDomain\MyUser After installation, you can add users or groups on the management console. However, global security groups and Active Directory Domain Services (AD DS) distribution groups are not supported. You must use **Domain local** or **Universal** groups to perform this action.
Website name	Specify the custom name that will be used to run the publishing service. If you do not have a custom name, you don't have to change it.
Port binding	Specify a unique port number that will be used by App-V. Example: **12345** Ensure that the port specified is not being used by another website.

12. On the **Configure Publishing Server Configuration** page, specify the following:

ITEM TO CONFIGURE	DESCRIPTION AND EXAMPLES
Specify the management service URL	Example: http://localhost:12345
Website name	Specify the custom website name that will be used to run the publishing service. If you do not have a custom name, do not make any changes.
Port binding	Specify a unique port number that will be used by App-V. Example: 54321 Ensure that the port specified is not being used by another website.

13. On the **Reporting Server** page, specify the following:

ITEM TO CONFIGURE	DESCRIPTION AND EXAMPLES
Website name	Specify the custom name that will be used to run the Reporting Service. If you do not have a custom name, do not make any changes.
Port binding	Specify a unique port number that will be used by App-V. Example: 55555 Ensure that the port specified is not being used by another website.

14. To start the installation, click **Install** on the **Ready** page, and then click **Close** on the **Finished** page.

15. To verify that the setup completed successfully, open a web browser, and type the following URL with the bracketed variables adjusted according to your specifications in the earlier steps:

    ```
    http://<Management server machine name>:<Management service port number>/console.html
    ```

    Example: `http://localhost:12345/console.html`. If the installation succeeded, the App-V Management console will display with no errors.

Related topics

- Deploying App-V
- How to install the management and reporting databases on separate computers from the management and reporting services
- How to install the publishing server on a remote computer
- How to deploy the App-V server using a script

How to deploy the App-V server using a script

5/31/2019 • 12 minutes to read • Edit Online

Applies to: Windows Server 2016

In order to complete the **appv_server_setup.exe** server setup successfully using the command line, you must specify and combine multiple parameters.

To install the App-V server using a script

Use the following lists and tables for more information about installing the App-V server using the command line.

The information in the following lists and tables can also be accessed through the command line by entering the following command: `appv\_server\_setup.exe /?`.

How to use common parameters to install the Management server and Management database on a local machine

The following examples will show you how to install the Management server and database on a local machine.

Parameters for a default instance of Microsoft SQL Server for a new installation on a local machine

To use the default instance of Microsoft SQL Server, use the following parameters:

- */MANAGEMENT_SERVER*
- */MANAGEMENT_ADMINACCOUNT*
- */MANAGEMENT_WEBSITE_NAME*
- */MANAGEMENT_WEBSITE_PORT*
- */DB_PREDEPLOY_MANAGEMENT*
- */MANAGEMENT_DB_SQLINSTANCE_USE_DEFAULT*
- */MANAGEMENT_DB_NAME*

Parameters for a custom instance of Microsoft SQL Server for a new installation on a local machine

To use a custom instance of Microsoft SQL Server, use the following parameters:

- */MANAGEMENT_SERVER*
- */MANAGEMENT_ADMINACCOUNT*
- */MANAGEMENT_WEBSITE_NAME*
- */MANAGEMENT_WEBSITE_PORT*
- */DB_PREDEPLOY_MANAGEMENT*
- */MANAGEMENT_DB_CUSTOM_SQLINSTANCE*
- */MANAGEMENT_DB_NAME*

Example parameters for using a custom instance of Microsoft SQL Server for a new installation on a local machine

```
/appv_server_setup.exe /QUIET
/MANAGEMENT_SERVER
/MANAGEMENT_ADMINACCOUNT="Domain\AdminGroup"
/MANAGEMENT_WEBSITE_NAME="Microsoft AppV Management Service"
/MANAGEMENT_WEBSITE_PORT="8080"
/DB_PREDEPLOY_MANAGEMENT
/MANAGEMENT_DB_CUSTOM_SQLINSTANCE="SqlInstanceName"
/MANAGEMENT_DB_NAME="AppVManagement"
```

How to use common parameters to install the Management server using an existing Management database on a local machine

The following examples will show you how to install the Management server on a local machine with an existing Management database.

Default instance of Microsoft SQL Server for installation with an existing Management database on a local machine

To use the default instance of Microsoft SQL Server, use the following parameters:

- */MANAGEMENT_SERVER*
- */MANAGEMENT_ADMINACCOUNT*
- */MANAGEMENT_WEBSITE_NAME*
- */MANAGEMENT_WEBSITE_PORT*
- */EXISTING_MANAGEMENT_DB_SQL_SERVER_USE_LOCAL*
- */EXISTING_MANAGEMENT_DB_SQLINSTANCE_USE_DEFAULT*
- */EXISTING_MANAGEMENT_DB_NAME*

Custom instance of Microsoft SQL Server for installation with an existing Management database on a local machine

To use a custom instance of Microsoft SQL Server, use these parameters:

- */MANAGEMENT_SERVER*
- */MANAGEMENT_ADMINACCOUNT*
- */MANAGEMENT_WEBSITE_NAME*
- */MANAGEMENT_WEBSITE_PORT*
- */EXISTING_MANAGEMENT_DB_SQL_SERVER_USE_LOCAL*
- */EXISTING_MANAGEMENT_DB_CUSTOM_SQLINSTANCE*
- */EXISTING_MANAGEMENT_DB_NAME*

Example parameters for using a custom instance of Microsoft SQL Server for installation with an existing Management database on a local machine

```
/appv_server_setup.exe /QUIET
/MANAGEMENT_SERVER
/MANAGEMENT_ADMINACCOUNT="Domain\AdminGroup"
/MANAGEMENT_WEBSITE_NAME="Microsoft AppV Management Service"
/MANAGEMENT_WEBSITE_PORT="8080"
/EXISTING_MANAGEMENT_DB_SQL_SERVER_USE_LOCAL
/EXISTING_MANAGEMENT_DB_CUSTOM_SQLINSTANCE ="SqlInstanceName"
/EXISTING_MANAGEMENT_DB_NAME ="AppVManagement"
```

How to install the Management server with an existing Management database on a remote machine

Default instance of Microsoft SQL Server with an existing Management database on a remote machine

To use the default instance of Microsoft SQL Server, use the following parameters:

- */MANAGEMENT_SERVER*
- */MANAGEMENT_ADMINACCOUNT*
- */MANAGEMENT_WEBSITE_NAME*
- */MANAGEMENT_WEBSITE_PORT*
- */EXISTING_MANAGEMENT_DB_REMOTE_SQL_SERVER_NAME*
- */EXISTING_MANAGEMENT_DB_SQLINSTANCE_USE_DEFAULT*
- */EXISTING_MANAGEMENT_DB_NAME*

Custom instance of Microsoft SQL Server with an existing Management database on a remote machine

To use a custom instance of Microsoft SQL Server, use these parameters:

- */MANAGEMENT_SERVER*
- */MANAGEMENT_ADMINACCOUNT*
- */MANAGEMENT_WEBSITE_NAME*
- */MANAGEMENT_WEBSITE_PORT*
- */EXISTING_MANAGEMENT_DB_REMOTE_SQL_SERVER_NAME*
- */EXISTING_MANAGEMENT_DB_CUSTOM_SQLINSTANCE*
- */EXISTING_MANAGEMENT_DB_NAME*

Example for using a custom instance of Microsoft SQL Server with an existing Management database on a remote machine

```
/appv_server_setup.exe /QUIET
/MANAGEMENT_SERVER
/MANAGEMENT_ADMINACCOUNT="Domain\AdminGroup"
/MANAGEMENT_WEBSITE_NAME="Microsoft AppV Management Service"
/MANAGEMENT_WEBSITE_PORT="8080"
/EXISTING_MANAGEMENT_DB_REMOTE_SQL_SERVER_NAME="SqlServermachine.domainName"
/EXISTING_MANAGEMENT_DB_CUSTOM_SQLINSTANCE ="SqlInstanceName"
/EXISTING_MANAGEMENT_DB_NAME ="AppVManagement"
```

Installing the Management database and the Management Server on the same computer

The following examples will show you how to install the Management server and database on the same computer.

Default instance of Microsoft SQL Server for installation on the same computer

To use the default instance of Microsoft SQL Server, use these parameters:

- */DB_PREDEPLOY_MANAGEMENT*
- */MANAGEMENT_DB_SQLINSTANCE_USE_DEFAULT*
- */MANAGEMENT_DB_NAME*
- */MANAGEMENT_SERVER_MACHINE_USE_LOCAL*
- */MANAGEMENT_SERVER_INSTALL_ADMIN_ACCOUNT*

Custom instance of Microsoft SQL Server for installation on the same computer

To use a custom instance of Microsoft SQL Server, use these parameters:

- */DB_PREDEPLOY_MANAGEMENT*
- */MANAGEMENT_DB_CUSTOM_SQLINSTANCE*
- */MANAGEMENT_DB_NAME*
- */MANAGEMENT_SERVER_MACHINE_USE_LOCAL*
- */MANAGEMENT_SERVER_INSTALL_ADMIN_ACCOUNT*

Example for using a custom instance of Microsoft SQL Server for installation on the same computer

```
/appv_server_setup.exe /QUIET
/DB_PREDEPLOY_MANAGEMENT
/MANAGEMENT_DB_CUSTOM_SQLINSTANCE="SqlInstanceName"
/MANAGEMENT_DB_NAME="AppVManagement"
/MANAGEMENT_SERVER_MACHINE_USE_LOCAL
/MANAGEMENT_SERVER_INSTALL_ADMIN_ACCOUNT="Domain\InstallAdminAccount"
```

Installing the Management database on a different computer than the Management server

The following examples will show you how to install the Management database and server on different computers.

Default instance of Microsoft SQL Server for installing the Management database on a different computer than the Management server

To use the default instance of Microsoft SQL Server, use the following parameters:

- */DB_PREDEPLOY_MANAGEMENT*
- */MANAGEMENT_DB_SQLINSTANCE_USE_DEFAULT*
- */MANAGEMENT_DB_NAME*
- */MANAGEMENT_REMOTE_SERVER_MACHINE_ACCOUNT*
- */MANAGEMENT_SERVER_INSTALL_ADMIN_ACCOUNT*

Custom instance of Microsoft SQL Server for installing the Management database on a different computer than the Management server

To use a custom instance of Microsoft SQL Server, use these parameters:

- */DB_PREDEPLOY_MANAGEMENT*
- */MANAGEMENT_DB_CUSTOM_SQLINSTANCE*
- */MANAGEMENT_DB_NAME*
- */MANAGEMENT_REMOTE_SERVER_MACHINE_ACCOUNT*
- */MANAGEMENT_SERVER_INSTALL_ADMIN_ACCOUNT*

Example for using a custom instance of Microsoft SQL Server for installing the Management database on a different computer than the Management server

```
/appv_server_setup.exe /QUIET
/DB_PREDEPLOY_MANAGEMENT
/MANAGEMENT_DB_CUSTOM_SQLINSTANCE="SqlInstanceName"
/MANAGEMENT_DB_NAME="AppVManagement"
/MANAGEMENT_REMOTE_SERVER_MACHINE_ACCOUNT="Domain\MachineAccount"
/MANAGEMENT_SERVER_INSTALL_ADMIN_ACCOUNT="Domain\InstallAdminAccount"
```

Installing the Publishing server

The following examples will show you how to install the Publishing server.

Default instance of Microsoft SQL Server for installing the Publishing server

To use the default instance of Microsoft SQL Server, use the following parameters:

- */PUBLISHING_SERVER*
- */PUBLISHING_MGT_SERVER*
- */PUBLISHING_WEBSITE_NAME*
- */PUBLISHING_WEBSITE_PORT*

Example for installing the Publishing server

```
/appv_server_setup.exe /QUIET
/PUBLISHING_SERVER
/PUBLISHING_MGT_SERVER="http://ManagementServerName:ManagementPort"
/PUBLISHING_WEBSITE_NAME="Microsoft AppV Publishing Service"
/PUBLISHING_WEBSITE_PORT="8081"
```

Installing the Reporting server and Reporting database on a local machine

The following examples will show you how to install the Reporting server and database on a local machine.

Default instance of Microsoft SQL Server for installing the Reporting server and Reporting database on a local machine

To use the default instance of Microsoft SQL Server, use the following parameters:

- */REPORTING_SERVER*
- */REPORTING_WEBSITE_NAME*
- */REPORTING_WEBSITE_PORT*
- */DB_PREDEPLOY_REPORTING*
- */REPORTING_DB_SQLINSTANCE_USE_DEFAULT*
- */REPORTING_DB_NAME*

Custom instance of Microsoft SQL Server for installing the Reporting server and Reporting database on a local machine

To use a custom instance of Microsoft SQL Server, use these parameters:

- */REPORTING_SERVER*
- */REPORTING_ADMINACCOUNT*
- */REPORTING_WEBSITE_NAME*
- */REPORTING_WEBSITE_PORT*
- */DB_PREDEPLOY_REPORTING*
- */REPORTING_DB_CUSTOM_SQLINSTANCE*
- */REPORTING_DB_NAME*

Example for using a custom instance of Microsoft SQL Server for installing the Reporting server and Reporting database on a local machine

```
/appv_server_setup.exe /QUIET
/REPORTING_SERVER
/REPORTING_WEBSITE_NAME="Microsoft AppV Reporting Service"
/REPORTING_WEBSITE_PORT="8082"
/DB_PREDEPLOY_REPORTING
/REPORTING_DB_CUSTOM_SQLINSTANCE="SqlInstanceName"
/REPORTING_DB_NAME="AppVReporting"
```

Installing the Reporting server using an existing Reporting database on a local machine

The following examples will show you how to install the reporting machine on a local machine with an existing Reporting database.

Default instance of Microsoft SQL Server for installing the Reporting server using an existing Reporting database on a local machine

To use the default instance of Microsoft SQL Server, use the following parameters:

- */REPORTING_SERVER*
- */REPORTING_WEBSITE_NAME*
- */REPORTING_WEBSITE_PORT*
- */EXISTING_REPORTING_DB_SQL_SERVER_USE_LOCAL*
- */EXISTING_REPORTING_DB_SQLINSTANCE_USE_DEFAULT*
- */EXISTING_REPORTING_DB_NAME*

Custom instance of Microsoft SQL Server for installing the Reporting server using an existing Reporting database on a local machine

To use a custom instance of Microsoft SQL Server, use these parameters:

- */REPORTING_SERVER*
- */REPORTING_ADMINACCOUNT*
- */REPORTING_WEBSITE_NAME*
- */REPORTING_WEBSITE_PORT*
- */EXISTING_REPORTING_DB_SQL_SERVER_USE_LOCAL*
- */EXISTING_REPORTING_DB_CUSTOM_SQLINSTANCE*
- */EXISTING_REPORTING_DB_NAME*

Example for using a custom instance of Microsoft SQL Server for installing the Reporting server using an existing Reporting database on a local machine

```
/appv_server_setup.exe /QUIET
/REPORTING_SERVER
/REPORTING_WEBSITE_NAME="Microsoft AppV Reporting Service"
/REPORTING_WEBSITE_PORT="8082"
/EXISTING_REPORTING_DB_SQL_SERVER_USE_LOCAL
/EXISTING_REPORTING _DB_CUSTOM_SQLINSTANCE="SqlInstanceName"
/EXITING_REPORTING_DB_NAME="AppVReporting"
```

Installing the Reporting server using an existing Reporting database on a remote machine

The following examples will show you how to install the Reporting server and on a remote machine with an existing database.

Default instance of Microsoft SQL Server for installing the Reporting server using an existing Reporting database on a remote machine

To use the default instance of Microsoft SQL Server, use the following parameters:

- */REPORTING_SERVER*
- */REPORTING_WEBSITE_NAME*
- */REPORTING_WEBSITE_PORT*
- */EXISTING_REPORTING_DB_REMOTE_SQL_SERVER_NAME*
- */EXISTING_REPORTING_DB_SQLINSTANCE_USE_DEFAULT*
- */EXISTING_REPORTING_DB_NAME*

Custom instance of Microsoft SQL Server for installing the Reporting server using an existing Reporting database on a remote machine

To use a custom instance of Microsoft SQL Server, use these parameters:

- */REPORTING_SERVER*
- */REPORTING_ADMINACCOUNT*
- */REPORTING_WEBSITE_NAME*
- */REPORTING_WEBSITE_PORT*

- */EXISTING_REPORTING_DB_REMOTE_SQL_SERVER_NAME*
- */EXISTING_REPORTING_DB_CUSTOM_SQLINSTANCE*
- */EXISTING_REPORTING_DB_NAME*

Example using a custom instance of Microsoft SQL Server for installing the Reporting server using an existing Reporting database on a remote machine

```
/appv_server_setup.exe /QUIET
/REPORTING_SERVER
/REPORTING_WEBSITE_NAME="Microsoft AppV Reporting Service"
/REPORTING_WEBSITE_PORT="8082"
/EXISTING_REPORTING_DB_REMOTE_SQL_SERVER_NAME="SqlServerMachine.DomainName"
/EXISTING_REPORTING _DB_CUSTOM_SQLINSTANCE="SqlInstanceName"
/EXITING_REPORTING_DB_NAME="AppVReporting"
```

Installing the Reporting database on the same computer as the Reporting server

The following examples will show you how to install the Reporting database and server on the same computer.

Default instance of Microsoft SQL Server for installing the Reporting database on the same computer as the Reporting server

To use the default instance of Microsoft SQL Server, use the following parameters:

- */DB_PREDEPLOY_REPORTING*
- */REPORTING_DB_SQLINSTANCE_USE_DEFAULT*
- */REPORTING_DB_NAME*
- */REPORTING_SERVER_MACHINE_USE_LOCAL*
- */REPORTING_SERVER_INSTALL_ADMIN_ACCOUNT*

Custom instance of Microsoft SQL Server for installing the Reporting database on the same computer as the Reporting server

To use a custom instance of Microsoft SQL Server, use these parameters:

- */DB_PREDEPLOY_REPORTING*
- */REPORTING_DB_CUSTOM_SQLINSTANCE*
- */REPORTING_DB_NAME*
- */REPORTING_SERVER_MACHINE_USE_LOCAL*
- */REPORTING_SERVER_INSTALL_ADMIN_ACCOUNT*

Example for using a custom instance of Microsoft SQL Server for installing the Reporting database on the same computer as the Reporting server

```
/appv_server_setup.exe /QUIET
/DB_PREDEPLOY_REPORTING
/REPORTING_DB_CUSTOM_SQLINSTANCE="SqlInstanceName"
/REPORTING_DB_NAME="AppVReporting"
/REPORTING_SERVER_MACHINE_USE_LOCAL
/REPORTING_SERVER_INSTALL_ADMIN_ACCOUNT="Domain\InstallAdminAccount"
```

Installing the Reporting database on a different computer than the Reporting server

The following examples will show you how to install the Reporting database and server on different computers.

Default instance of Microsoft SQL Server for installing the Reporting database on a different computer than the Reporting server

To use the default instance of Microsoft SQL Server, use the following parameters:

- */DB_PREDEPLOY_REPORTING*
- */REPORTING_DB_SQLINSTANCE_USE_DEFAULT*
- */REPORTING_DB_NAME*
- */REPORTING_REMOTE_SERVER_MACHINE_ACCOUNT*
- */REPORTING_SERVER_INSTALL_ADMIN_ACCOUNT*

Custom instance of Microsoft SQL Server for installing the Reporting database on a different computer than the Reporting server

To use a custom instance of Microsoft SQL Server, use these parameters:

- */DB_PREDEPLOY_REPORTING*
- */REPORTING_DB_CUSTOM_SQLINSTANCE*
- */REPORTING_DB_NAME*
- */REPORTING_REMOTE_SERVER_MACHINE_ACCOUNT*
- */REPORTING_SERVER_INSTALL_ADMIN_ACCOUNT*

Example for using a custom instance of Microsoft SQL Server for installing the Reporting database on a different computer than the Reporting server

```
/appv_server_setup.exe /QUIET
/DB_PREDEPLOY_REPORTING
/REPORTING_DB_CUSTOM_SQLINSTANCE="SqlInstanceName"
/REPORTING_DB_NAME="AppVReporting"
/REPORTING_REMOTE_SERVER_MACHINE_ACCOUNT="Domain\MachineAccount"
/REPORTING_SERVER_INSTALL_ADMIN_ACCOUNT="Domain\InstallAdminAccount"
```

Parameter definitions

- General parameters
- Management Server installation parameters
- Management Server Database parameters
- Publishing Server installation parameters
- Reporting Server parameters
- Parameters for using an existing Reporting Server database
- Reporting Server database installation parameters
- Parameters for using an existing Management Server database

Parameter definitions for general parameters

PARAMETER	DESCRIPTION
/QUIET	Specifies silent install.
/UNINSTALL	Specifies an uninstall.
/LAYOUT	Specifies layout action. This extracts the MSIs and script files to a folder without installing the actual product. No value is expected.
/LAYOUTDIR	Specifies the layout directory with a string. For example, `/LAYOUTDIR="C:\Application Virtualization Server"`.

PARAMETER	DESCRIPTION
/INSTALLDIR	Specifies the installation directory with a string. For example, `/INSTALLDIR="C:\Program Files\Application Virtualization\Server"` .
/MUOPTIN	Enables Microsoft Update. No value is expected.
/ACCEPTEULA	Accepts the license agreement. This is required for an unattended installation. For example, `/ACCEPTEULA` or `/ACCEPTEULA=1` .

Parameter definitions for Management Server installation parameters

PARAMETER	DESCRIPTION
/MANAGEMENT_SERVER	Specifies that the management server will be installed. No value is expected.
/MANAGEMENT_ADMINACCOUNT	Specifies the account that will be allowed administrator access to the management server. This account can be an individual user account or a group. For example, `/MANAGEMENT_ADMINACCOUNT="mydomain\admin"` . If **/MANAGEMENT_SERVER** isn't specified, this parameter will be ignored.
/MANAGEMENT_WEBSITE_NAME	Specifies name of the website that will be created for the management service. For example, `/MANAGEMENT_WEBSITE_NAME="Microsoft App-V Management Service"`
/MANAGEMENT_WEBSITE_PORT	Specifies the port number that will be used by the management service will use. For example, `/MANAGEMENT_WEBSITE_PORT=82` .

Parameter definitions for the Management Server Database

PARAMETER	DESCRIPTION
/DB_PREDEPLOY_MANAGEMENT	Specifies that the management database will be installed. You must have sufficient database permissions to complete this installation. No value is expected.
/MANAGEMENT_DB_SQLINSTANCE_USE_DEFAULT	Indicates that the default SQL instance should be used. No value is expected.
/MANAGEMENT_DB_CUSTOM_SQLINSTANCE	Specifies the name of the custom SQL instance that should be used to create a new database. For example, `/MANAGEMENT_DB_CUSTOM_SQLINSTANCE="MYSQLSERVER"` . If **/DB_PREDEPLOY_MANAGEMENT** isn't specified, this parameter will be ignored.
/MANAGEMENT_DB_NAME	Specifies the name of the new management database that should be created. For example, `/MANAGEMENT_DB_NAME="AppVMgmtDB"` . If **/DB_PREDEPLOY_MANAGEMENT** isn't specified, this will be ignored.

PARAMETER	DESCRIPTION
/MANAGEMENT_SERVER_MACHINE_USE_LOCAL	Indicates if the management server that will be accessing the database is installed on the local server. This is a switch parameter, so no value is expected.
/MANAGEMENT_REMOTE_SERVER_MACHINE_ACCOUNT	Specifies the machine account of the remote machine that the management server will be installed on. For example, `/MANAGEMENT_REMOTE_SERVER_MACHINE_ACCOUNT="domain\computername"` .
/MANAGEMENT_SERVER_INSTALL_ADMIN_ACCOUNT	Indicates the Administrator account that will be used to install the management server. For example, `/MANAGEMENT_SERVER_INSTALL_ADMIN_ACCOUNT ="domain\alias"` .

Parameter definitions for Publishing Server installation parameters

PARAMETER	DESCRIPTION
/PUBLISHING_SERVER	Specifies that the publishing server will be installed. No value is expected.
/PUBLISHING_MGT_SERVER	Specifies the URL to Management Service the Publishing server will connect to. For example, `http://<management server name>;<Management server port number>` . If **/PUBLISHING_SERVER** isn't used, this parameter will be ignored.
/PUBLISHING_WEBSITE_NAME	Specifies name of the website that will be created for the publishing service. For example, `/PUBLISHING_WEBSITE_NAME="Microsoft App-V Publishing Service"` .
/PUBLISHING_WEBSITE_PORT	Specifies the port number used by the publishing service. For example, `/PUBLISHING_WEBSITE_PORT=83` .

Parameter definitions for Reporting Server

PARAMETER	DESCRIPTION
/REPORTING_SERVER	Specifies that the Reporting Server will be installed. No value is expected.
/REPORTING_WEBSITE_NAME	Specifies name of the website that will be created for the Reporting Service. For example, `/REPORTING_WEBSITE_NAME="Microsoft App-V ReportingService"` .
/REPORTING_WEBSITE_PORT	Specifies the port number that the Reporting Service will use. For example, `/REPORTING_WEBSITE_PORT=82` .

Parameters for using an existing Reporting Server database

PARAMETER	DESCRIPTION
/EXISTING_REPORTING_DB_SQL_SERVER_USE_LOCAL	Indicates that the Microsoft SQL Server is installed on the local server. This is a switch parameter, so no value is expected.
/EXISTING_REPORTING_DB_REMOTE_SQL_SERVER_NAME	Specifies the name of the remote computer that SQL Server is installed on. Takes a string. For example, `/EXISTING_REPORTING_DB_REMOTE_SQL_SERVER_NAME="mycomputer1"` .
/EXISTING_REPORTING_DB_SQLINSTANCE_USE_DEFAULT	Indicates that the default SQL instance is to be used. This is a switch parameter, so no value is expected.
/EXISTING_REPORTING_DB_CUSTOM_SQLINSTANCE	Specifies the name of the custom SQL instance that should be used. Takes a string. For example, `/EXISTING_REPORTING_DB_CUSTOM_SQLINSTANCE="MYSQLSERVER"` .
/EXISTING_REPORTING_DB_NAME	Specifies the name of the existing Reporting database that should be used. Takes a string. For example, `/EXISTING_REPORTING_DB_NAME="AppVReporting"` .

Parameter definitions for Reporting Server database installation

PARAMETER	DESCRIPTION
/DB_PREDEPLOY_REPORTING	Specifies that the Reporting Database will be installed. DBA permissions are required for this installation. No value is expected.
/REPORTING_DB_SQLINSTANCE_USE_DEFAULT	Specifies the name of the custom SQL instance that should be used. Takes a string. For example, `/REPORTING_DB_CUSTOM_SQLINSTANCE="MYSQLSERVER"` .
/REPORTING_DB_NAME	Specifies the name of the new Reporting database that should be created. Takes a string. For example, `/REPORTING_DB_NAME="AppVMgmtDB"` .
/REPORTING_SERVER_MACHINE_USE_LOCAL	Indicates that the Reporting server that will be accessing the database is installed on the local server. This is a switch parameter, so no value is expected.
/REPORTING_REMOTE_SERVER_MACHINE_ACCOUNT	Specifies the machine account of the remote machine that the Reporting server will be installed on. Takes a string. For example, `/REPORTING_REMOTE_SERVER_MACHINE_ACCOUNT = "domain\computername"` .
/REPORTING_SERVER_INSTALL_ADMIN_ACCOUNT	Indicates the Administrator account that will be used to install the App-V Reporting Server. Takes a string. For example, `/REPORTING_SERVER_INSTALL_ADMIN_ACCOUNT = "domain\alias"` .

Parameters for using an existing Management Server database

PARAMETER	DESCRIPTION
/EXISTING_MANAGEMENT_DB_SQL_SERVER_USE_LOCAL	Indicates that the SQL Server is installed on the local server. Switch parameter so no value is expected. If **/DB_PREDEPLOY_MANAGEMENT** isn't specified, this variable will be ignored.
/EXISTING_MANAGEMENT_DB_REMOTE_SQL_SERVER_NAME	Specifies the name of the remote computer that SQL Server is installed on. Takes a string. For example, `/EXISTING_MANAGEMENT_DB_REMOTE_SQL_SERVER_NAME="mycomputer1"` .
/EXISTING_MANAGEMENT_DB_SQLINSTANCE_USE_DEFAULT	Indicates that the default SQL instance is to be used. Switch parameter so no value is expected. If **/DB_PREDEPLOY_MANAGEMENT** isn't specified, this variable will be ignored.
/EXISTING_MANAGEMENT_DB_CUSTOM_SQLINSTANCE	Specifies the name of the custom SQL instance that will be used. For example, `/EXISTING_MANAGEMENT_DB_CUSTOM_SQLINSTANCE="AppVManagement"` . If **/DB_PREDEPLOY_MANAGEMENT** isn't specified, this will be ignored.
/EXISTING_MANAGEMENT_DB_NAME	Specifies the name of the existing management database that should be used. For example, `/EXISTING_MANAGEMENT_DB_NAME="AppVMgmtDB"` . If **/DB_PREDEPLOY_MANAGEMENT** isn't specified, this will be ignored.

Related topics

- Deploying the App-V Server

How to deploy the App-V databases by using SQL scripts

5/31/2019 • 4 minutes to read • Edit Online

Applies to: Windows Server 2016

Use the following instructions to use SQL scripts, rather than the Windows Installer, to:

- Install the App-V databases
- Upgrade the App-V databases to a later version

> **NOTE**
> If you have already deployed an App-V 5.0 SP3 database or later, the SQL scripts are not required to upgrade to App-V for Windows.

How to install the App-V databases by using SQL scripts

1. Before you install the database scripts, review and keep a copy of the App-V license terms. By running the database scripts, you are agreeing to the license terms. If you do not accept them, you should not use this software.

2. Copy **appv_server_setup.exe** from the App-V release media to a temporary location.

3. From a command prompt, run **appv_server_setup.exe** and specify a temporary location for extracting the database scripts.

```
appv\_server\_setup.exe /layout c:\\_<temporary location path>_
```

4. Browse to the temporary location that you created, open the extracted **DatabaseScripts** folder, and review the appropriate **readme.txt** file for instructions:

DATABASE	LOCATION OF README.TXT FILE TO USE
Management database	ManagementDatabase subfolder
Reporting database	ReportingDatabase subfolder

Caution

The readme.txt file in the ManagementDatabase subfolder is out of date. The information in the updated readme files below is the most current and should supersede the readme information provided in the **DatabaseScripts** folders.

> **IMPORTANT**
> The InsertVersionInfo.sql script is not required for versions of the App-V management database later than App-V 5.0 SP3. The Permissions.sql script should be updated according to Step 2 in KB article 3031340. Step 1 is not required for versions of App-V later than App-V 5.0 SP3.

Updated management database README file content

```
*****************************************************************************************************
Before you install and use the Application Virtualization Database Scripts, you must:

- Review the license terms.
- Print and retain a copy of the license terms for your records.

By running the App-V you agree to such license terms. If you do not accept them, do not use the software.
*****************************************************************************************************

Steps to install "AppVManagement" schema in SQL SERVER.

## PREREQUISITES:

 1. Review the installation package.  The following files MUST exist:

    SQL files
    ---------
    Database.sql
    CreateTables.sql
    CreateStoredProcs.sql
    UpdateTables.sql
    Permissions.sql

 2. Ensure the target SQL Server instance and SQL Server Agent service are running.

 3. If you are not running the scripts directly on the server, ensure the
    necessary SQL Server client software is installed and available from
    the specified location.  Specifically, the "osql" command must be supported for these scripts to run.

## PREPARATION:

 1. Review the database.sql file and modify as necessary.  Although the
    defaults are likely sufficient, it is suggested that the following
    settings be reviewed:

    DATABASE - ensure name is satisfactory - default is "AppVManagement".

 2. Review the Permissions.sql file and provide all the necessary account information
    for setting up read and write access on the database. Note: Default settings in the file will not work.

## INSTALLATION:

 1. Run the database.sql against the "master" database.  Your user
    credential must have the ability to create databases.
    This script will create the database.

 2. Run the following scripts against the "AppVManagement" database using the
    same account as above in order.

    CreateTables.sql
    CreateStoredProcs.sql
    UpdateTables.sql
    Permissions.sql
```

Updated reporting database README file content

```
*****************************************************************************************************
Before you install and use the Application Virtualization Database Scripts, you must:

- Review the license terms.
- Print and retain a copy of the license terms for your records.

By running the App-V you agree to such license terms. If you do not accept them, do not use the software.
*****************************************************************************************************
```

```
*****************************************************************************************************

Steps to install "AppVReporting" schema in SQL SERVER.

## PREREQUISITES:

 1. Review the installation package.  The following files MUST exist:

    SQL files
    ---------
    Database.sql
    UpgradeDatabase.sql
    CreateTables.sql
    CreateReportingStoredProcs.sql
    CreateStoredProcs.sql
    CreateViews.sql
    Permissions.sql
    ScheduleReportingJob.sql

 2. Ensure the target SQL Server instance and SQL Server Agent service are running.

 3. If you are not running the scripts directly on the server, ensure the
    necessary SQL Server client software is installed and executable from
    the location you have chosen.  Specifically, the "osql" command must be supported for these scripts to
run.

## PREPARATION:

 1. Review the database.sql file and modify as necessary.  Although the
    defaults are likely sufficient, it is suggested that the following
    settings be reviewed:

    DATABASE - ensure name is satisfactory - default is "AppVReporting".

 2. Review the Permissions.sql file and provide all the necessary account information
    for setting up read and write access on the database. Note: Default settings
    in the file will not work.

 3. Review the ScheduleReportingJob.sql file and make sure that the stored proc schedule
    time is acceptable. The default stored proc schedule time is at 12.01 AM (line 84).
    If this time is not suitable, you can change this to a more suitable time. The time is in the format
HHMMSS.

## INSTALLATION:

 1. Run the database.sql against the "master" database.  Your user
    credential must have the ability to create databases.
    This script will create the database.

 2. If upgrading the database, run UpgradeDatabase.sql This will upgrade database schema.

 2. Run the following scripts against the "AppVReporting" database using the
    same account as above in order.

    CreateTables.sql
    CreateReportingStoredProcs.sql
    CreateStoredProcs.sql
    CreateViews.sql
    Permissions.sql
    ScheduleReportingJob.sql
```

Related topics

- Deploying the App-V Server
- How to deploy the App-V Server

How to install the publishing server on a remote computer

6/6/2019 • 2 minutes to read • Edit Online

Applies to: Windows Server 2016

Use the following procedure to install the publishing server on a separate computer. Before you perform the following procedure, ensure the database and management server are available.

Installing the publishing server on a separate computer

1. Copy the App-V server installation files to the computer on which you want to install it on. To start the App-V server installation, run **appv_server_setup.exe** as an administrator, then select **Install**.
2. On the **Getting started** page, review and accept the license terms, then select **Next**.
3. On the **Use Microsoft Update to help keep your computer secure and up-to-date** page, to enable Microsoft updates, select **Use Microsoft Update when I check for updates (recommended)**. To disable Microsoft Update, select **I don't want to use Microsoft Update**. Click **Next**.
4. On the **Feature selection** page, select the **Publishing Server** checkbox, then select **Next**.
5. On the **Installation location** page, accept the default location, then select **Next**.
6. On the **Configure publishing server configuration** page, specify the following items:
   - The URL for the management service that the publishing server will connect to. For example, **http://ManagementServerName:12345**.
   - Specify the website name that you want to use for the publishing service. If you don't have a custom name, then use the default name.
   - For the **Port binding**, specify a unique port number that will be used by App-V. For example, **54321**.
7. On the **Ready to install** page, select **Install**.
8. After the installation is complete, the publishing server must be registered with the management server. In the App-V management console, use the following steps to register the server:
   a. Open the App-V management server console.
   b. In the left pane, select **Servers**, then select **Register New Server**.
   c. Enter the server name and a description (if required), then select **Add**.
9. To verify that the publishing server is running correctly, you should import a package to the management server, entitle that package to an AD group, then publish it. Using an internet browser, open the following URL: **https://publishingserver:pubport**. If the server is running correctly, information like the following example should appear.

```
<Publishing Protocol="1.0">

    <Packages>

    <Package PackageId="28115343-06e2-44dc-a327-3a0b9b868bda" VersionId="5d03c08f-51dc-4026-8cf9-
15ebe3d65a72" PackageUrl="\\server\share\file.appv" />

    </Packages>

    <NoGroup>

    <Package PackageId="28115343-06e2-44dc-a327-3a0b9b868bda" />

    </NoGroup>

</Publishing>
```

Related topics

- Deploying App-V

How to Install the Management and Reporting Databases on separate computers from the Management and Reporting Services

6/10/2019 • 3 minutes to read • Edit Online

Applies to: Windows Server 2016

Use the following procedure to install the database server and management server on different computers. The computer you plan to install the database server on must be running a supported version of Microsoft SQL or the installation will fail.

> **NOTE**
> After you complete the deployment, the administrator installing the service will need the Microsoft SQL Server name, instance name and the database name to connect to these databases.

Installing the management database and the management server on separate computers

1. Copy the App-V server installation files to the computer you want to install it on. To start the App-V server installation, run **appv_server_setup.exe** as an administrator, then select **Install**.
2. On the **Getting started** page, review and accept the license terms, then select **Next**.
3. On the **Use Microsoft Update to help keep your computer secure and up-to-date** page, to enable Microsoft updates, select **Use Microsoft Update when I check for updates (recommended).** To disable Microsoft updates, select **I don't want to use Microsoft Update**, then select **Next**.
4. On the **Feature selection** page, select the components you want to install by first selecting the **Management Server Database** checkbox, then selecting **Next**.
5. On the **Installation location** page, accept the default location and select **Next**.
6. On the initial **Create new management server database** page, accept the default selections if appropriate, then select **Next**.
   - If you are using a custom SQL Server instance, select **Use a custom instance** and enter the name of the instance.
   - If you are using a custom database name, select **Custom configuration** and enter the database name.
7. On the next **Create new management server database** page, select **Use a remote computer**, then enter the remote machine account using the following format: `Domain\MachineAccount`.

   > **NOTE**
   > If you plan to deploy the management server on the same computer you must select **Use this local computer**. Specify the user name for the management server **Install Administrator** using the following format: `Domain\AdministratorLoginName`. After that, select **Next**.

8. To start the installation, select **Install**.

Installing the reporting database and the reporting server on separate computers

1. Copy the App-V server installation files to the computer you want to install it on. To start the App-V server installation, run **appv_server_setup.exe** as an administrator, then select **Install**.

2. On the **Getting started** page, review and accept the license terms, then select **Next**.

3. On the **Use Microsoft Update to help keep your computer secure and up-to-date** page, to enable Microsoft Update, select **Use Microsoft Update when I check for updates (recommended)**. To disable Microsoft Update, select **I don't want to use Microsoft Update**. After that, select **Next**.

4. On the **Feature selection** page, select the components you want to install by first selecting the **Reporting Server Database** checkbox, then selecting **Next**.

5. On the **Installation Location** page, accept the default location and select **Next**.

6. On the initial **Create new management server database** page, accept the default selections if appropriate, then select **Next**.

   - If you're using a custom SQL Server instance, select **Use a custom instance** and enter the instance name.
   - If you're using a custom database name, select **Custom configuration** and enter the database name.

7. On the next **Create new management server database** page, select **Use a remote computer**, and enter the remote machine account using the following format: `Domain\MachineAccount`.

   > **NOTE**
   > If you plan to deploy the reporting server on the same computer you must select **Use this local computer**. Specify the user name for the reporting server **Install Administrator** using the following format: Domain\AdministratorLoginName. After that, select **Next**.

8. To start the installation, select **Install**.

Installing the management and reporting databases using App-V database scripts

1. Copy the App-V server installation files to the computer on which you want to install it on.

2. To extract the App-V database scripts, open a command prompt and specify the location where the installation files are saved and run the following command:

   ```
   appv\_server\_setup.exe /LAYOUT /LAYOUTDIR="InstallationExtractionLocation"
   ```

3. After the extraction has been completed, to access the App-V database scripts and instructions readme file:

   - The App-V Management Database scripts and instructions readme are located in the following folder: **InstallationExtractionLocation** \ **Database Scripts** \ **Management Database**.
   - The App-V Reporting Database scripts and instructions readme are located in the following folder: **InstallationExtractionLocation** \ **Database Scripts** \ **Reporting Database**.

4. For each database, copy the scripts to a share and modify them following the instructions in the readme file.

> **NOTE**
>
> For more information about modifying the required SIDs contained in the scripts see, How to Install the App-V Databases and Convert the Associated Security Identifiers by Using Windows PowerShell. 5. Run the scripts on the computer running Microsoft SQL Server.

Related topics

- Deploying App-V

How to install the Management Server on a Standalone Computer and Connect it to the Database

6/10/2019 • 2 minutes to read • Edit Online

Applies to: Windows Server 2016

To install the management server on a standalone computer and connect it to the database, follow these steps.

1. Copy the App-V server installation files to the computer on which you want to install it on. To start the App-V server installation, run **appv_server_setup.exe** as an administrator, then select **Install**.
2. On the **Getting Started** page, review and accept the license terms, then select **Next**.
3. On the **Use Microsoft Update to help keep your computer secure and up-to-date** page, to enable Microsoft Update, select **Use Microsoft Update when I check for updates (recommended)**. To disable Microsoft Update, select **I don't want to use Microsoft Update**, then select **Next**.
4. On the **Feature Selection** page, select the **Management Server** checkbox, then select **Next**.
5. On the **Installation Location** page, accept the default location, then select **Next**.
6. On the **Configure Existing Management Database** page, select **Use a remote SQL Server**, then enter the computer running Microsoft SQL's machine name, such as `SqlServerMachine`.

> **NOTE**
>
> If the Microsoft SQL Server is deployed on the same server, select **Use local SQL Server**. For the SQL Server Instance, select **Use the default instance**. If you are using a custom Microsoft SQL Server instance, you must select **Use a custom instance**, then enter the instance's name. Specify the **SQL Server Database name** that this management server will use, such as `AppvManagement`. 7. On the **Configure management server configuration** page, specify the following items:

- The AD group or account that will connect to the management console for administrative purposes for example **MyDomain\MyUser** or **MyDomain\AdminGroup**. The account or AD group you specify will be enabled to manage the server through the management console. You can add additional users or groups using the management console after installation
- The **Website Name** you want to use for the management service. Accept the default if you do not have a custom name.
- For the **Port Binding**, specify a unique port number, such as **12345**.

8. Select **Install**.
9. To confirm that the setup has completed successfully, open a web browser and enter the following URL: https://managementserver:portnumber/Console. If the installation was successful, you should see the **Management Console** appear without any error messages or warnings displayed.

Related topics

- Deploying App-V

About App-V reporting

6/10/2019 • 9 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Application Virtualization (App-V) includes a built-in reporting feature that collects information about computers running the App-V client and virtual application package usage. You can generate reports from a centralized database with this information.

App-V reporting overview

The following list displays the end–to-end high-level workflow for reporting in App-V.

1. The App-V Reporting server requires the following things:

   - Internet Information Service (IIS) web server role
   - Windows Authentication role (under **IIS / Security**)
   - SQL Server installed and running with SQL Server Reporting Services (SSRS)

     To confirm SQL Server Reporting Services is running, enter https://localhost/Reports in a web browser as administrator on the server that will host App-V Reporting. The SQL Server Reporting Services Home page should appear.

2. Install the App-V reporting server and associated database. For more information about installing the reporting server see How to install the Reporting Server on a standalone computer and connect it to the database. Configure the time when the computer running the App-V client should send data to the reporting server.

3. If you are not using an electronic software distribution system such as Configuration Manager to view reports then you can define reports in SQL Server Reporting Service. Download predefined appvshort Reports from the Download Center at Application Virtualization SSRS Reports.

   > **NOTE**
   >
   > If you are using the Configuration Manager integration with App-V, most reports are generated from Configuration Manager rather than from App-V.

4. After importing the App-V Windows PowerShell module using **Import-Module AppvClient** as administrator, enable App-V client reporting. This sample Windows PowerShell command enables App-V reporting:

   ```
   Set-AppvClientConfiguration -ReportingServerURL <url>:<port> -ReportingEnabled 1 -ReportingStartTime
   <0-23> -ReportingRandomDelay <#min>
   ```

   To immediately send App-V report data, run **Send-AppvClientReport** on the App-V client.

   For more information about configuring reporting on the App-V client, see About client configuration settings. To administer App-V Reporting with Windows PowerShell, see How to enable reporting on the App-V client by using PowerShell.

5. After the reporting server receives the data from the App-V client it sends the data to the reporting database. When the database receives and processes the client data, a successful reply is sent to the reporting server, which then notifies the App-V client.

6. When the App-V client receives the success notification, it empties the data cache to conserve space.

> **NOTE**
>
> By default, the cache is cleared after the server confirms receipt of data. You can manually configure the client to save the data cache.

If the App-V client device does not receive a success notification from the server, it retains data in the cache and tries to resend data at the next configured interval. Clients continue to collect data and add it to the cache.

App-V reporting server frequently asked questions

The following sections provide answers to frequently asked questions about how App-V reporting works.

How frequently is reporting information sent to the reporting database?

Frequency depends on the computer running the App-V client's reporting configuration. You must configure the frequency or interval for sending the reporting data manually, as App-V reporting is not enabled by default.

What information is stored in the reporting server database?

The following information is stored in the reporting database:

- The operating system running on the computer running the App-V client: host name, version, service pack, type - client/server, processor architecture.
- App-V client information: version.
- Published package list: GUID, version GUID, name.
- Application usage information: name, version, streaming server, user (domain\alias), package version GUID, launch status and time, shutdown time.

What is the average volume of information that is sent to the reporting server?

It depends. Three sets of data can be sent to the reporting server:

- Operating system and App-V client information, which is about 150 Bytes every time it gets sent to the server.
- Published package lists, which are about 7 KB per 30 packages. This is sent only when the package list is updated with a publishing refresh, which is done infrequently; if there is no change, this information is not sent.
- Virtual application usage information is about 0.25 KB per event. Opening and closing count as one event if both occur before sending the information. When sending using a scheduled task, only the data since the last successful upload is sent to the server. If sending manually through the Windows PowerShell cmdlet, there is an optional argument called **DeleteOnSuccess** that controls if the data needs to be re-sent the next time around.

For example, if twenty applications are opened and closed and reporting information is scheduled to be sent daily, the typical daily traffic should be about 0.15 KB + 20 × 0.25 KB, or about 5 KB/user.

Can I schedule reporting?

Yes. Besides manually sending reporting using Windows PowerShell cmdlets (**Send-AppvClientReport**), the task can be scheduled so it will happen automatically. There are two ways to schedule the reporting:

- Using a Windows PowerShell cmdlet: **Set-AppvClientConfiguration**. For example:

```
Set-AppvClientConfiguration -ReportingEnabled 1 -ReportingServerURL http://any.com/appv-reporting
```

For a complete list of client configuration settings, go to About client configuration settings and look for the following entries: **ReportingEnabled**, **ReportingServerURL**, **ReportingDataCacheLimit**, **ReportingDataBlockSize**, **ReportingStartTime**, **ReportingRandomDelay**, **ReportingInterval**.

- Using Group Policy. If distributed using the domain controller, the settings are the same as previously listed.

> **NOTE**
> Group Policy settings override local settings configured using Windows PowerShell.

App-V Client reporting

To use App-V reporting you must enable and configure the App-V client. To configure reporting on the client, use the Windows PowerShell cmdlet **Set-AppVClientConfiguration**, or the Group Policy **ADMX Template**. For more information about the Windows PowerShell cmdlets, see About client configuration settings. The following section provides examples of Windows PowerShell commands for configuring App-V client reporting.

Configuring App-V client reporting using Windows PowerShell

The following examples show how Windows PowerShell parameters can configure the reporting features of the App-V client.

> **NOTE**
> The following configuration tasks can also be configured using Group Policy settings in the App-V ADMX template. The App-V settings are under **Computer Configuration** > **Administrative Templates** > **System** > **App-V**.

Enabling reporting and initiating data collection on the computer running the App-V client

Use the following command to enable reporting and initiate date collection on the computer running the App-V client:

```
Set-AppVClientConfiguration –ReportingEnabled 1
```

Configuring the client to automatically send data to a specific reporting server

To configure the client to automatically send data to a specific reporting server, use a command with the following format:

```
Set-AppVClientConfiguration -ReportingServerURL http://MyReportingServer:MyPort/ -ReportingStartTime 20 -
ReportingInterval 1 -ReportingRandomDelay 30
```

This example command configures the client to automatically send the reporting data to the reporting server URL `http://MyReportingServer:MyPort/`. The reporting data will be sent to the URL daily between 8:00 PM and 8:30 PM, depending on the session's generated random delay.

Limiting the size of the client's data cache

To limit the client's data cache size, use a command with the following format:

```
Set-AppvClientConfiguration –ReportingDataCacheLimit 100
```

This example command configures the maximum size of the App-V client computer's reporting cache to 100 MB. If the cache limit is reached before the data is sent to the server, then the log rolls over and data will be overwritten as necessary.

Configuring the data block size transmitted across the network between the client and the server

To configure the data block size, use a command with the following format:

```
Set-AppvClientConfiguration –ReportingDataBlockSize 10240
```

This example command specifies the maximum size of the data block as 10,240 MB.

Types of data collected

The following table displays the types of information you can collect by using App-V reporting.

CLIENT INFORMATION	PACKAGE INFORMATION	APPLICATION USAGE
Host name	Package name	Start and end times
App-V client version	Package version	Run status
Processor architecture	Package source	Shutdown state
Operating system version	Percent cached	Application name
Service Pack level		Application version
Operating system type		Username
		Connection group

The client collects and saves this data in an **.xml** format. The data cache is hidden by default and requires administrator rights to open the XML file.

Sending data to the server

You can configure the computer that is running the App-V client to automatically send data to the specified reporting server. To specify the server, use the **Set-AppvClientConfiguration** cmdlet with the following settings:

- ReportingEnabled
- ReportingServerURL
- ReportingStartTime
- ReportingInterval
- ReportingRandomDelay

After you configure the previous settings, you must create a scheduled task. The scheduled task will contact the server specified by the **ReportingServerURL** setting and will initiate the transfer. If you want to manually send data outside of the scheduled times, use the following Windows PowerShell cmdlet:

```
Send-AppVClientReport –URL http://MyReportingServer:MyPort/ -DeleteOnSuccess
```

If the reporting server has been previously configured, then the **–URL** parameter can be omitted. Alternatively, if the data should be sent to an alternate location, specify a different URL to override the configured **ReportingServerURL** for this data collection.

The **-DeleteOnSuccess** parameter indicates that if the transfer is successful, then the data cache will be cleared. If this is not specified, then the cache will not be cleared.

Manual Data Collection

You can also use the **Send-AppVClientReport** cmdlet to manually collect data. This solution is helpful with or

without an existing reporting server. The following list displays information about collecting data with or without a reporting server.

WITH A REPORTING SERVER	WITHOUT A REPORTING SERVER
f you have an existing App-V reporting server, create a customized scheduled task or script. Specify that the client sends the data to the specified location at the desired frequency.	If you do not have an existing App-V reporting Server, use the **–URL** parameter to send the data to a specified share. For example: `Send-AppVClientReport –URL \\Myshare\MyData\ -DeleteOnSuccess` The previous example will send the reporting data to the `\\MyShare\MyData\` location indicated by the **-URL** parameter. After the data has been sent, the cache is cleared.

> **NOTE**
>
> If a location other than the Reporting Server is specified, the data is sent in **.xml** format with no additional processing.

Creating reports

To retrieve report information and create reports using App-V you must use one of the following methods:

- Microsoft SQL Server Reporting Services (SSRS)—Microsoft SSRS is available with Microsoft SQL Server. SSRS is not installed when you install the App-V reporting server. It must be deployed separately to generate the associated reports. For more information, see the What is SQL Server Reporting Services (SSRS)? article.
- Scripting—You can generate reports by scripting directly against the App-V reporting database. For example:

  Stored Procedure:

  spProcessClientReport is scheduled to run at midnight or 12:00 AM.

  To run the Microsoft SQL Server Scheduled Stored procedure, the Microsoft SQL Server Agent must be running. Make sure the Microsoft SQL Server Agent is set to **AutoStart**. For more information, see Autostart SQL Server Agent (SQL Server Management Studio).

  The stored procedure is also created when you use the App-V database scripts.

You should also ensure that the reporting server web service's **Maximum Concurrent Connections** is set to a value that the server can manage without affecting availability. The recommended number of **Maximum Concurrent Connections** for the **Reporting Web Service** is **10,000**.

Related topics

- Deploying the App-V server
- How to install the reporting server on a standalone computer and connect it to the database

How to install the reporting server on a standalone computer and connect it to the database

6/10/2019 • 2 minutes to read • Edit Online

Applies to: Windows Server 2016

Use the following procedure to install the reporting server on a standalone computer and connect it to the database.

> **IMPORTANT**
> Before performing the following procedure you should read and understand About App-V reporting.

Install the reporting server on a standalone computer and connect it to the database

1. Copy the App-V server installation files to the computer you plan to install it on. To start the App-V server installation, run **appv_server_setup.exe** as an administrator, then select **Install**.
2. On the **Getting started** page, review and accept the license terms, then select **Next**.
3. On the **Use Microsoft Update to help keep your computer secure and up-to-date** page, to enable Microsoft Update, select **Use Microsoft Update when I check for updates (recommended)**. To disable Microsoft Update, select **I don't want to use Microsoft Update**. Select **Next**.
4. On the **Feature selection** page, select the **Reporting Server** checkbox, then select **Next**.
5. On the **Installation location** page, accept the default location and select **Next**.
6. On the **Configure existing reporting database** page, select **Use a remote SQL Server**, then enter the machine name of the computer running Microsoft SQL Server. For example, you can name your computer **SqlServerMachine**.

   > **NOTE**
   > If the Microsoft SQL Server is deployed on the same server, select **Use local SQL Server**. For the SQL Server instance, select **Use the default instance**. If you're using a custom Microsoft SQL Server instance, select **Use a custom instance**, then enter the name of your custom instance. Specify the **SQL Server Database name** that this reporting server will use; for example, you can name the server **AppvReporting**.

7. On the **Configure reporting server configuration** page.
   - Specify the website name you want to use for the reporting service. Leave the default unchanged if you do not have a custom name.
   - For the **Port binding**, specify a unique, five-digit port number for App-V to use, such as **55555**. Make sure that the specified port isn't being used by another website.
8. Select **Install**.

Related topics

- About App-V reporting
- Deploying App-V
- How to enable reporting on the App-V client by using Windows PowerShell

App-V Deployment Checklist

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

This checklist outlines the recommended steps and items to consider when deploying App-V features. Use it to organize your priorities while you deploy App-V. You can copy this checklist into a spreadsheet program and customize it for your use.

STATUS	TASK	REFERENCES	NOTES
☐	Prepare the computing environment for App-V deployment during your planning phase.	App-V planning checklist	
☐	Review App-V's supported configurations.	App-V supported configurations	
☐	Run App-V Setup to deploy the required App-V features for your environment.	How to install the sequencer Enable the App-V desktop client How to deploy the App-V server	

NOTE

Keep track of server names and associated URLs you create during installation. You'll need this information throughout the installation process.

Related topics

- Deploying App-V

Deploying Microsoft Office 2016 by using App-V

6/26/2019 • 18 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Use the information in this article to use Application Virtualization (App-V) to deliver Microsoft Office 2016 as a virtualized application to computers in your organization. For information about using App-V to deliver Office 2013, see Deploying Microsoft Office 2013 by using App-V. For information about using App-V to deliver Office 2010, see Deploying Microsoft Office 2010 by using App-V.

What to know before you start

Before you deploy Office 2016 with App-V, review the following planning information.

Supported Office versions and Office coexistence

Use the following table to get information about supported versions of Office and running coexisting versions of Office.

INFORMATION TO REVIEW	DESCRIPTION
Supported versions of Microsoft Office	Supported versions of Office and deployment types (for example, desktop, personal Virtual Desktop Infrastructure (VDI), and pooled VDI), and Office licensing options.
Planning for using App-V with coexisting versions of Office	Considerations for installing different versions of Office on the same computer.

Packaging, publishing, and deployment requirements

Before you deploy Office with App-V, review the following requirements.

TASK	REQUIREMENT
Packaging.	All Office applications that you deploy to users must be in a single package. In App-V 5.0 and later, you must use the Office Deployment Tool to create packages. The Sequencer doesn't support package creation. If you're deploying Microsoft Visio 2016 and Microsoft Project 2016 at the same time as Office, you must put them all in the same package. For more information, see Deploying Visio 2016 and Project 2016 with Office.
Publishing.	You can only publish one Office package per client computer. You must publish the Office package globally, not to the user.
Deploying Office 365 ProPlus, Visio Pro for Office 365, or Project Pro for Office 365 to a shared computer with Remote Desktop Services.	You must enable shared computer activation.

Excluding Office applications from a package

The following table describes the recommended methods for excluding specific Office applications from a

package.

TASK	DETAILS
Use the **ExcludeApp** setting when you create the package by using the Office Deployment Tool.	With this setting, you can exclude specific Office applications from the package that the Office Deployment Tool creates. For example, you can use this setting to create a package that contains only Microsoft Word. For more information, see ExcludeApp element.
Modify the DeploymentConfig.xml file	Modify the DeploymentConfig.xml file after the package has been created. This file contains the default package settings for all users on a computer that is running the App-V Client. For more information, see Disabling Office 2016 applications.

Creating an Office 2016 package for App-V with the Office Deployment Tool

Complete the following steps to create an Office 2016 package for App-V.

> **IMPORTANT**
>
> In App-V 5.0 and later, you must use the Office Deployment Tool to create a package. You cannot use the Sequencer to create packages.

Review prerequisites for using the Office Deployment Tool

The computer on which you are installing the Office Deployment Tool must have the following:

PREREQUISITE	DESCRIPTION
Prerequisite software	.Net Framework 4
Supported operating systems	64-bit version of Windows 10 64-bit version of Windows 8 or 8.1 64-bit version of Windows 7

> **NOTE**
>
> In this topic, the term "Office 2016 App-V package" refers to subscription licensing.

Create Office 2016 App-V packages with the Office Deployment Tool

You create Office 2016 App-V packages by using the Office Deployment Tool. The following instructions explain how to create an Office 2016 App-V package with subscription licensing.

Create Office 2016 App-V packages on 64-bit Windows computers. Once created, the Office 2016 App-V package will run on 32-bit and 64-bit Windows 7, Windows 8.1, and Windows 10 computers.

Download the Office Deployment Tool

Office 2016 App-V packages are created using the Office Deployment Tool, which generates an Office 2016 App-V Package. The package cannot be created or modified through the App-V sequencer. To begin package creation, follow these steps:

1. Download the Office 2016 Deployment Tool for Click-to-Run.

> **IMPORTANT**
> You must use the Office 2016 Deployment Tool to create Office 2016 App-V Packages.

2. Run the .exe file and extract its features into the desired location. To make this process easier, you can create a shared network folder where the features will be saved.

   Example location: \\Server\Office2016

3. Check that the **setup.exe** and **configuration.xml** files exist and are in the location you specified.

Download Office 2016 applications

After you download the Office Deployment Tool, you can use it to get the latest Office 2016 applications. After getting the Office applications, you create the Office 2016 App-V package.

The XML file included in the Office Deployment Tool specifies the product details, such as the languages and Office applications included.

1. **Customize the sample XML configuration file:** Use the sample XML configuration file that you downloaded with the Office Deployment Tool to customize the Office applications:

   a. Open the sample XML file in Notepad or your favorite text editor.

   b. With the sample **configuration.xml** file open and ready for editing, you can specify products, languages, and the path to which you save the Office 2016 applications. The following is a basic example of the **configuration.xml** file:

```
<Configuration>
   <Add SourcePath= ”\\Server\Office2016” OfficeClientEdition="32" >
    <Product ID="O365ProPlusRetail ">
      <Language ID="en-us" />
    </Product>
    <Product ID="VisioProRetail">
      <Language ID="en-us" />
    </Product>
  </Add>
</Configuration>
```

> **NOTE**
> The configuration XML is a sample XML file. The file includes lines that are commented out. You can “uncomment” these lines to customize additional settings with the file. To uncomment these lines, remove the `<! - -` from the beginning of the line, and the `-- >` from the end of the line.

The previous example of an XML configuration file specifies that Office 2016 ProPlus 32-bit edition, including Visio ProPlus, will be downloaded in English to the \\server\Office2016 location where Office applications will be saved. Note that the Product ID of the applications will not affect Office's final licensing. You can create Office 2016 App-V packages with various licensing from the same applications by specifying licensing in a later stage. The following table summarizes the XML file's customizable attributes and elements:

INPUT	DESCRIPTION	EXAMPLE
Add element	Specifies which products and languages the package will include.	N/A

INPUT	DESCRIPTION	EXAMPLE
OfficeClientEdition (attribute of **Add** element)	Specifies whether Office 2016 32-bit or 64-bit edition will be used. **OfficeClientEdition** must be set to a valid value for the operation to succeed.	`OfficeClientEdition="32"` `OfficeClientEdition="64"`

Product element	Specifies the application. Project 2016 and Visio 2016 must be specified here as added products to include them in the applications.
	For more information about Product IDs, see Product IDs that are supported by the Office Deployment Tool for Click-to-Run.
		`Product ID ="O365ProPlusRetail"`
		`Product ID ="VisioProRetail"`
		`Product ID ="ProjectProRetail"`

Language element	Specifies which language the applications support.	`Language ID="en-us"`
Version (attribute of **Add** element)	Optional. Specifies which build the package will use. Defaults to latest advertised build (as defined in v32.CAB at the Office source).	`16.1.2.3`
SourcePath (attribute of **Add** element)	Specifies the location the applications will be saved to.	`Sourcepath = "\\Server\Office2016"`

Channel (part of **Add** element)	Optional. Defines which channel will be used to update Office after installation.
	The default is **Deferred** for Office 365 ProPlus and **Current** for Visio Pro for Office 365 and Project Desktop Client.
	For more information about update channels, see Overview of update channels for Office 365 ProPlus.
		`Channel="Current"`
		`Channel="Deferred"`
		`Channel="FirstReleaseDeferred"`
		`Channel="FirstReleaseCurrent"`

After editing the **configuration.xml** file to specify the desired product, languages, and the location where the Office 2016 applications will be saved to, you can save the configuration file under a name of your choice, such as "Customconfig.xml." 2. **Download the applications into the specified location:** Use an elevated command prompt and a 64-bit operating system to download the Office 2016 applications that will later be converted into an App-V package. The following is an example command:

```
\\server\Office2016\setup.exe /download \\server\Office2016\Customconfig.xml
```

The following table describes the example command's elements:

ELEMENT	DESCRIPTION
`\\server\Office2016`	This is the network share location that contains the Office Deployment Tool and the custom **Configuration.xml** file, which in this example is **Customconfig.xml**.
`Setup.exe`	This is the Office Deployment Tool.

ELEMENT	DESCRIPTION
`/download`	Downloads the Office 2016 applications that you specify in the **Customconfig.xml** file.
`\\server\Office2016\Customconfig.xml`	This passes the XML configuration file required to complete the download process. In this example, the file used is **Customconfig.xml**. After using the download command, Office applications should be found in the location specified in the configuration file, which in this example is `\\Server\Office2016`.

Convert the Office applications into an App-V package

After you download the Office 2016 applications through the Office Deployment Tool, use the Office Deployment Tool to convert them into an Office 2016 App-V package. Complete the steps that correspond to your licensing model.

What you'll need to do

- Create the Office 2016 App-V packages on 64-bit Windows computers. However, the package will run on 32-bit and 64-bit Windows 7, Windows 8 or 8.1, and Windows 10 computers.
- Create an Office App-V package for either Subscription Licensing package by using the Office Deployment Tool, and then modify the **Customconfig.xml** configuration file.

  The following table summarizes the values you need to enter in the **Customconfig.xml** file. The steps in the sections that follow the table will specify the exact entries you need to make.

> **NOTE**
> You can use the Office Deployment Tool to create App-V packages for Office 365 ProPlus. Creating packages for the volume-licensed versions of Office Professional Plus or Office Standard is not supported.

PRODUCT ID	SUBSCRIPTION LICENSING
Office 2016	O365ProPlusRetail
Office 2016 with Visio 2016	O365ProPlusRetail VisioProRetail
Office 2016 with Visio 2016 and Project 2016	O365ProPlusRetail VisioProRetail ProjectProRetail

How to convert the Office applications into an App-V package

1. In Notepad, reopen the CustomConfig.xml file, and make the following changes to the file:
   - **SourcePath**: Change to the location where you saved the Office applications you downloaded during setup.
   - **ProductID**: Specify the type of licensing, as shown in the following example:
     - Subscription Licensing:

```
<Configuration>
<Add SourcePath= "\\server\Office 2016" OfficeClientEdition="32" >
  <Product ID="O365ProPlusRetail">
    <Language ID="en-us" />
  </Product>
  <Product ID="VisioProRetail">
    <Language ID="en-us" />
  </Product>
</Add>
</Configuration>
```

This example made the following changes to create this Subscription Licensing package:

- **SourcePath** was changed to point to the Office applications that were downloaded earlier.
- **Product ID** for Office was changed to `O365ProPlusRetail`.
- **Product ID** for Visio was changed to `VisioProRetail`.

- **ExcludeApp** (optional): Lets you specify Office programs that you don't want included in the App-V package created by the Office Deployment Tool. For example, you can exclude Access.
- **PACKAGEGUID** (optional): By default, all App-V packages created by the Office Deployment Tool share the same App-V Package ID. You can use **PACKAGEGUID** to specify a different package ID for each package, which allows you to publish multiple App-V packages created by the Office Deployment Tool, and then manage your published packages with the App-V Server.

  An example of when to use this parameter is if you create different packages for different users. For example, you can create a package with just Office 2016 for some users, and create another package with Office 2016 and Visio 2016 for another set of users.

  > **NOTE**
  > Even if you use unique package IDs, you can still deploy only one App-V package to a single device.

2. Use the /packager command to convert the Office applications to an Office 2016 App-V package.

   The following is an example packager command:

```
\\server\Office2016\setup.exe /packager \\server\Office2016\Customconfig.xml
\\server\share\Office2016AppV
```

The following table describes each element used in the example command:

ELEMENT	DESCRIPTION
`\\server\Office2016`	This is the network share location that contains the Office Deployment Tool and the custom Configuration.xml file, which in this example is Customconfig.xml.
`Setup.exe`	This is the Office Deployment Tool.
`/packager`	This command creates the Office 2016 App-V package with the license type specified in the Customconfig.xml file.

ELEMENT	DESCRIPTION
`\\server\Office2016\Customconfig.xml`	This passes the configuration XML file that has been prepared for the packaging stage. In this example, the file is Customconfig.xml.
`\\server\share\Office2016AppV`	This specifies the location of the newly created Office App-V package.

After you run the **/packager** command, the following folders appear up in the directory where you specified the package should be saved:

- **App-V Packages**—contains an Office 2016 App-V package and two deployment configuration files.
- **WorkingDir**

> **NOTE**
> To troubleshoot any issues, see the log files in the %temp% directory (default).

3. Verify that the Office 2016 App-V package works correctly:
   a. Publish the Office 2016 App-V package that you created globally to a test computer and verify that the Office 2016 shortcuts appear.
   b. Start a few Office 2016 applications, such as Excel or Word, to ensure that your package is working as expected.

Publishing the Office package for App-V

Use the following information to publish an Office package.

Methods for publishing Office App-V packages

Deploy the App-V package for Office 2016 by using the same methods as the other packages that you've already deployed:

- System Center Configuration Manager
- App-V Server
- Stand-alone through Windows PowerShell commands

Publishing prerequisites and requirements

PREREQUISITE OR REQUIREMENT	DETAILS
Enable Windows PowerShell scripting on the App-V clients.	To publish Office 2016 packages, you must run a script. However, package scripts are disabled by default on App-V clients. To enable scripting, run the following Windows PowerShell command: `Set-AppvClientConfiguration -EnablePackageScripts 1`
Publish the Office 2016 package globally.	Extension points in the Office App-V package require installation at the computer level. When you publish at the computer level, no prerequisite actions or redistributables are needed. The Office 2016 package globally enables its applications to work like natively installed Office, eliminating the need for administrators to customize packages.

How to publish an Office package

Run the following command to publish an Office package globally:

```
Add-AppvClientPackage <Path_to_AppV_Package > | Publish-AppvClientPackage -global
```

- You can add permissions to a group of computers instead of just a user group through the Web Management Console on the App-V Server. This lets packages be published globally to the computers in the corresponding group.

Customizing and managing Office App-V packages

To manage your Office App-V packages, use the same operations as you would for any other package, with a few exceptions as outlined in the following sections.

- Enabling Office plug-ins by using connection groups
- Disabling Office 2016 applications
- Disabling Office 2016 shortcuts
- Managing Office 2016 package upgrades
- Deploying Visio 2016 and Project 2016 with Office

Enabling Office plug-ins by using connection groups

The following steps will tell you how to enable Office plug-ins with your Office package. To use Office plug-ins, you must use the App-V Sequencer to create a separate package that only contains the plug-ins (you can't use the Office Deployment Tool to create the plug-ins package). After that, create a connection group that contains the Office package and the plug-ins package.

Enable plug-ins for Office App-V packages

1. Add a Connection Group through App-V Server, System Center Configuration Manager, or a Windows PowerShell cmdlet.

2. Sequence your plug-ins using the App-V Sequencer. Ensure that Office 2016 is installed on the computer that will be used to sequence the plug-in. We recommend that you use Office 365 ProPlus (non-virtual) on the sequencing computer when sequencing Office 2016 plug-ins.

3. Create an App-V package that includes the plug-ins you want.

4. Add a Connection Group through the App-V Server, System Center Configuration Manager, or a Windows PowerShell cmdlet.

5. Add the Office 2016 App-V package and the plug-ins package you sequenced to the Connection Group you created.

   > **IMPORTANT**
   >
   > The order of the packages in the Connection Group determines the order in which the package contents are merged. In your Connection group descriptor file, add the Office 2016 App-V package first, and then add the plug-in App-V package.

6. Ensure that both packages are published to the target computer and that the plug-in package is published globally to match published Office 2016 App-V package's global settings.

7. Verify that the plug-in package's Deployment Configuration file has the same settings as the Office 2016 App-V package.

   The Office 2016 App-V plug-in package's settings must match those of the operating system to allow for

integration. You can search the Deployment Configuration File for “COM Mode” and ensure that your plug-ins package has that value set as “Integrated” and that both "InProcessEnabled" and "OutOfProcessEnabled" match the settings of the Office 2016 App-V package you published.

8. Open the Deployment Configuration File and set the value for **Objects Enabled** to **false**.
9. If you made any changes to the Deployment Configuration file after sequencing, ensure that the plug-in package is published with the file.
10. Ensure that the Connection Group you created on your desired computer is enabled. The Connection Group created will be shown as "pending" if the Office 2016 App-V package is being used while the Connection Group is enabled. If that happens, you'll have to reboot the computer to successfully enable the Connection Group.
11. After you successfully publish both packages and enable the Connection Group, verify the plug-ins you published on the Connection Group work as expected.

Disabling Office 2016 applications

You can also disable specific applications in your Office App-V package. For example, if you don't want to use Access, you can disable Access while leaving all other Office applications available. When you disable an application, the user will no longer see its shortcut icon. You don't need to re-sequence the application to do this. When you change the Deployment Configuration File after the Office 2016 App-V package has been published, just save the changes and add the Office 2016 App-V package, then republish it with the new Deployment Configuration File to apply the new settings to Office 2016 App-V Package applications.

> **NOTE**
> To exclude specific Office applications when you create the App-V package with the Office Deployment Tool, use the **ExcludeApp** setting.

Disable an Office 2016 application

1. Open a Deployment Configuration File with a text editor such as **Notepad** and search for “Applications."
2. Search for the Office application you want to disable, for example, Access 2016.
3. Change the value of **Enabled** to **false**.
4. Save the Deployment Configuration File.
5. Add the Office 2016 App-V Package with the new Deployment Configuration File.

```
<Application Id="[{AppVPackageRoot}]\office16\lync.exe" Enabled="true">
  <VisualElements>
    <Name>Lync 2016</Name>
    <Icon />
    <Description />
  </VisualElements>
</Application>
<Application Id="[(AppVPackageRoot}]\office16\MSACCESS.EXE" Enabled="true">
  <VisualElements>
    <Name>Access 2016</Name>
    <Icon />
    <Description />
  </VisualElements>
</Application>
```

6. Re-add the Office 2016 App-V package, and then republish it with the new Deployment Configuration File to apply the new settings to Office 2016 App-V Package applications.

Disabling Office 2016 shortcuts

You may want to disable shortcuts for certain Office applications instead of unpublishing or removing the package. The following example shows how to disable shortcuts for Microsoft Access.

Disable shortcuts for Office 2016 applications

1. Open a Deployment Configuration File in Notepad and search for "Shortcuts".

2. To disable certain shortcuts, delete or comment out the specific shortcuts you don't want. You must keep the subsystem present and enabled. The following example shows how to delete the Microsoft Access shortcuts while keeping the subsystems `<shortcut> </shortcut>` intact.

```
Shortcuts

-->
 <Shortcuts Enabled="true">
  <Extensions>
    <Extension Category="AppV.Shortcut">
      <Shortcut>
       <File>[{Common Programs}]\Microsoft Office 2016\Access 2016.lnk</File>
       <Target>[{AppvPackageRoot}])office16\MSACCESS.EXE</Target>
       <Icon>[{Windows}]\Installer\{90150000-000F-0000-0000-000000FF1CE)\accicons.exe.0.ico</Icon>
       <Arguments />
       <WorkingDirectory />
       <AppuserModelId>Microsoft.Office.MSACCESS.EXE.16</AppUserModelId>
       <AppUserModelExcludeFromShowInNewInstall>true</AppUserModelExcludeFromShowInNewInstall>
       <Description>Build a professional app quickly to manage data.</Description>
       <ShowCommand>1</ShowCommand>
       <ApplicationId>[{AppVPackageRoot}]\office16\MSACCESS.EXE</ApplicationId>
    </Shortcut>
```

3. Save the Deployment Configuration File.

4. Republish the Office 2016 App-V Package with the new Deployment Configuration File.

Many additional settings can be changed through modifying the Deployment Configuration for App-V packages, for example, file type associations, Virtual File System, and more. For additional information on how to use Deployment Configuration Files to change App-V package settings, refer to the additional resources section at the end of this document.

Managing Office 2016 package upgrades

To upgrade an Office 2016 package, use the Office Deployment Tool. To upgrade a previously deployed Office 2016 package, perform the steps outlined in the following section.

How to upgrade a previously deployed Office 2016 package

1. Create a new Office 2016 package through the Office Deployment Tool that uses the most recent Office 2016 application software. The most recent Office 2016 bits can always be obtained through the download stage of creating an Office 2016 App-V Package. The newly created Office 2016 package will have the most recent updates and a new Version ID. All packages created using the Office Deployment Tool have the same lineage.

> **NOTE**
>
> Office App-V packages have two Version IDs:
>
> - An Office 2016 App-V Package Version ID that is unique across all packages created using the Office Deployment Tool.
> - A second App-V Package Version ID, formatted as X.X.X.X, in the AppX manifest that will only change if there is a new version of Office itself. For example, if a new Office 2016 release with upgrades is available, and a package is created through the Office Deployment Tool to incorporate these upgrades, the X.X.X.X version ID will change to reflect the new version of Office. The App-V server will use the X.X.X.X version ID to differentiate this package and recognize that it contains new upgrades to the previously published package, and as a result, publish it as an upgrade to the existing Office 2016 package.

2. Globally publish the newly created Office 2016 App-V Packages onto the computers you want to apply the new updates to. Since the new package has the same lineage as the older Office 2016 App-V Package, publishing will be fast, as publishing the new package with the updates only applies the new changes to the old package.

3. Upgrades will be applied in the same manner of any globally published App-V packages. Because applications will probably be in use, upgrades might be delayed until the computer is rebooted.

Deploying Visio 2016 and Project 2016 with Office

The following table describes the requirements and options for deploying Visio 2016 and Project 2016 with Office.

TASK	DETAILS
How do I package and publish Visio 2016 and Project 2016 with Office?	You must include Visio 2016 and Project 2016 in the same package with Office. If you are not deploying Office, you can create a package that contains Visio and/or Project, as long as you follow the packaging, publishing, and deployment requirements described in this topic.
How can I deploy Visio 2016 and Project 2016 to specific users?	Use one of the following methods: **To create two different packages and deploy each one to a different group of users**: Create and deploy the following packages: - A package that contains only Office—deploy to computers whose users need only Office. - A package that contains Office, Visio, and Project—deploy to computers whose users need all three applications. **To create only one package for the whole organization, or to create a package intended for users who share computers**: 1. Create a package that contains Office, Visio, and Project. 2. Deploy the package to all users. 3. Use AppLocker to prevent specific users from using Visio and Project.

Related topics

- Deploying App-V for Windows 10
- Deploying Microsoft Office 2013 by using App-V
- Deploying Microsoft Office 2010 by using App-V
- Office 2016 Deployment Tool for Click-to-Run

Deploying Microsoft Office 2013 by Using App-V

6/26/2019 • 19 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Use the information in this article to use Application Virtualization (App-V) to deliver Microsoft Office 2013 as a virtualized application to computers in your organization. For information about using App-V to deliver Office 2010, see Deploying Microsoft Office 2010 by Using App-V. To successfully deploy Office 2013 with App-V, you need to be familiar with Office 2013 and App-V.

What to know before you start

Before you deploy Office 2013 with App-V, review the following planning information.

Supported Office versions and Office coexistence

The following table will direct you to more information about which versions of Office App-V supports and how to run App-V with coexisting versions of Office.

INFORMATION TO REVIEW	DESCRIPTION
Supported versions of Microsoft Office	- Supported versions of Office - Supported deployment types like desktop, personal Virtual Desktop Infrastructure (VDI), and pooled VDI - Office licensing options.
Planning for using App-V with coexisting versions of Office	Considerations for installing different versions of Office on the same computer.

Packaging, publishing, and deployment requirements

Before you deploy Office with App-V, review the following requirements.

TASK	REQUIREMENT
Packaging	All Office applications you wish to deploy to users must be in a single package. In App-V and later, you must use the Office Deployment Tool to create packages. The Sequencer doesn't support package creation. If you're deploying Microsoft Visio 2013 and Microsoft Project 2013 along with Office, you must include them in the same package with Office. For more information, see Deploying Visio 2013 and Project 2013 with Office.
Publishing	You can only publish one Office package per client computer. You must publish the Office package globally, not to the user.
Deploying Office 365 ProPlus, Visio Pro for Office 365, or Project Pro for Office 365 to a shared computer using Remote Desktop Services.	You must enable shared computer activation. You don't need to use shared computer activation if you're deploying a volume licensed product, such as Office Professional Plus 2013, Visio Professional 2013, or Project Professional 2013.

Excluding Office applications from a package

The following table describes the recommended methods for excluding specific Office applications from a package.

TASK	DETAILS
Use the **ExcludeApp** setting when you create the package by using the Office Deployment Tool.	Enables you to exclude specific Office applications from the package when the Office Deployment Tool creates the package. For example, you can use this setting to create a package that contains only Microsoft Word. For more information, see ExcludeApp element.
Modify the **DeploymentConfig.xml** file	Modify the **DeploymentConfig.xml** file after creating the package. This file contains the default package settings for all users on a computer running the App-V Client. For more information, see Disabling Office 2013 applications.

Creating an Office 2013 package for App-V with the Office Deployment Tool

Complete the following steps to create an Office 2013 package for App-V or later.

> **IMPORTANT**
>
> In App-V and later, you must the Office Deployment Tool to create a package. You cannot use the Sequencer to create packages.

Review prerequisites for using the Office Deployment Tool

Before you start, make sure that the computer on which you are installing the Office Deployment Tool has the following:

PREREQUISITE	DESCRIPTION
Prerequisite software	.NET Framework 4
Supported operating systems	64-bit version of Windows 8 or later 64-bit version of Windows 7

> **NOTE**
>
> In this topic, the term "Office 2013 App-V package" refers to subscription licensing and volume licensing.

Create Office 2013 App-V packages using Office Deployment Tool

You create Office 2013 App-V packages with the Office Deployment Tool. The following instructions explain how to create an Office 2013 App-V package with Volume Licensing or Subscription Licensing.

Create Office 2013 App-V packages on 64-bit Windows computers. Once created, the Office 2013 App-V package will run on 32-bit and 64-bit Windows 7, Windows 8.1, and Windows 10 computers.

Download the Office Deployment Tool

Office 2013 App-V Packages are created using the Office Deployment Tool, which generates an Office 2013 App-V Package. The App-V sequencer can't create or modify packages. To create a package:

1. Download the Office 2013 Deployment Tool for Click-to-Run.
2. Run the .exe file and extract its features into the desired location. To make this process easier, you can create a shared network folder where the features will be saved.

   Example: \\Server\Office2013

3. Check that a **setup.exe** and a **configuration.xml** file exist and are in the location you specified.

Download Office 2013 applications

After you download the Office Deployment Tool, you can use it to get the latest Office 2013 applications. You can create the Office 2013 App-V package after getting all the Office applications.

The XML file included in the Office Deployment Tool specifies the product details, such as the languages and Office applications included.

1. **Customize the sample XML configuration file:** Use the sample XML configuration file that you downloaded with the Office Deployment Tool to customize the Office applications:

   a. Open the sample XML file in Notepad or your favorite text editor.

   b. With the sample configuration.xml file open and ready for editing, you can specify products, languages, and the path to which you save the Office 2013 applications. The following is a basic example of the configuration.xml file:

```
<Configuration>
   <Add SourcePath= ”\\Server\Office2013” OfficeClientEdition="32" >
    <Product ID="O365ProPlusRetail ">
      <Language ID="en-us" />
    </Product>
    <Product ID="VisioProRetail">
      <Language ID="en-us" />
    </Product>
  </Add>
</Configuration>
```

> **NOTE**
>
> The configuration XML is a sample XML file. This file includes lines that are commented out. You can “uncomment” these lines to customize additional settings with the file.

   The previous example of an XML configuration file specifies that Office 2013 ProPlus 32-bit edition, including Visio ProPlus, will be downloaded in English to the \\server\Office 2013, which is the location where Office applications will be saved to. Note that the Product ID of the applications will not affect the final licensing of Office. Office 2013 App-V packages with various licensing can be created from the same applications by specifying licensing in a later stage. For more information, see Customizable attributes and elements of the XML file, later in this topic.

   After editing the configuration.xml file to specify the desired product, languages, and also the location which the Office 2013 applications will be saved onto, you can save the configuration file, for example, as Customconfig.xml.

2. **Download the applications into the specified location:** Use an elevated command prompt and a 64-bit operating system to download the Office 2013 applications that will later be converted into an App-V package. The following is an example command:

```
\\server\Office2013\setup.exe /download \\server\Office2013\Customconfig.xml
```

The following is a table that describes each element of the command:

ELEMENT	DESCRIPTION
`\\server\Office2013`	This is the network share location that contains the Office Deployment Tool and the custom Configuration.xml file, **Customconfig.xml**.
`setup.exe`	This is the Office Deployment Tool.
`/download`	This downloads the Office 2013 applications that you specify in the **Customconfig.xml** file. These bits can be later converted in an Office 2013 App-V package with Volume Licensing.
`\\server\Office2013\Customconfig.xml`	This passes the XML configuration file required to complete the download process. In this example, the file used is **Customconfig.xml**. After using the download command, Office applications should be found in the location specified in the XML configuration file, which in this example is `\\Server\Office2013`.

Customizable attributes and elements of the XML file

INPUT AND DESCRIPTION	EXAMPLE
Add element: Specifies the products and languages to include in the package.	N/A
OfficeClientEdition (attribute of Add element): Specifies the edition of Office 2013 product to use: 32-bit or 64-bit. The operation fails if **OfficeClientEdition** is not set to a valid value.	`OfficeClientEdition="32"` `OfficeClientEdition="64"`

Product element:
Specifies the application. Project 2013 and Visio 2013 must be specified here as an added product to be included in the applications.
	`Product ID="O365ProPlusRetail"`
	`Product ID="VisioProRetail"`
	`Product ID="ProjectProRetail"`
	`Product ID="ProPlusVolume"`
	`Product ID="ProjectProVolume"`

Language element: Specifies the language supported in the applications.	`Language ID="en-us"`
Version (attribute of Add element): Optional. Specifies a build to use for the package. Defaults to latest advertised build (as defined in v32.CAB at the Office source).	`15.1.2.3`
SourcePath (attribute of Add element): Specifies the location where the applications will be saved to.	`Sourcepath="\Server\Office2013”`

Convert the Office applications into an App-V package

After you download the Office 2013 applications through the Office Deployment Tool, use the Office Deployment

Tool to convert them into an Office 2013 App-V package. You'll need to make sure to have the right procedure depending on your licensing model.

What you'll need to do

- Create the Office 2013 App-V packages on 64-bit Windows computers. However, the package will run on 32-bit and 64-bit Windows 7, Windows 8, and Windows 10 computers.
- Create an Office App-V package for either the Subscription Licensing package or Volume Licensing by using the Office Deployment Tool, then modify the **Customconfig.xml** configuration file.

  The following table summarizes the values you need to enter in the **Customconfig.xml** file for the licensing model you're using. The steps in the sections that follow the table will specify the exact entries you need to make.

PRODUCT ID	VOLUME LICENSING	SUBSCRIPTION LICENSING
Office 2013	ProPlusVolume	O365ProPlusRetail
Office 2013 with Visio 2013	ProPlusVolume VisioProVolume	O365ProPlusRetail VisioProRetail
Office 2013 with Visio 2013 and Project 2013	ProPlusVolume VisioProVolume ProjectProVolume	O365ProPlusRetail VisioProRetail ProjectProRetail

How to convert the Office applications into an App-V package

1. In Notepad, reopen the CustomConfig.xml file, and make the following changes to the file:
   - **SourcePath**: Point to the Office applications downloaded earlier.
   - **ProductID**: Specify the type of licensing, as shown in the following examples:
     - Subscription Licensing:

       ```
       <Configuration>
       <Add SourcePath= "\\server\Office 2013" OfficeClientEdition="32" >
         <Product ID="O365ProPlusRetail">
           <Language ID="en-us" />
         </Product>
         <Product ID="VisioProRetail">
           <Language ID="en-us" />
         </Product>
       </Add>
       </Configuration>
       ```

       In this example, the following changes were made to create a package with Subscription licensing:

     - **SourcePath** is the path, which was changed to point to the Office applications that were downloaded earlier.
     - **Product ID** for Office was changed to `O365ProPlusRetail`.
     - **Product ID** for Visio was changed to `VisioProRetail`.
     - Volume Licensing

```
<Configuration>
<Add SourcePath= "\\Server\Office2013" OfficeClientEdition="32" >
  <Product ID="ProPlusVolume">
    <Language ID="en-us" />
  </Product>
  <Product ID="VisioProVolume">
    <Language ID="en-us" />
  </Product>
</Add>
</Configuration>
```

In this example, the following changes were made to create a package with Volume licensing:

- **SourcePath** is the source's path, which was changed to point to the Office applications that were downloaded earlier.
- **Product ID** for Office was changed to `ProPlusVolume`.
- **Product ID** for Visio was changed to `VisioProVolume`.

- **ExcludeApp** (optional) lets you specify Office programs that you don't want included in the App-V package that the Office Deployment Tool creates. For example, you can exclude Access and InfoPath.
- **PACKAGEGUID** (optional)—By default, all App-V packages created by the Office Deployment Tool share the same App-V Package ID. You can use PACKAGEGUID to specify a different package ID for each package, which allows you to publish multiple App-V packages, created by the Office Deployment Tool, and manage them by using the App-V Server.

  An example of when to use this parameter is if you create different packages for different users. For example, you can create a package with just Office 2013 for some users, and create another package with Office 2013 and Visio 2013 for another set of users.

  > **NOTE**
  > Even if you use unique package IDs, you can still deploy only one App-V package to a single device.

2. Use the **/packager** command to convert the Office applications to an Office 2013 App-V package.

   For example:

```
\\server\Office2013\setup.exe /packager \\server\Office2013\Customconfig.xml
\\server\share\Office2013AppV
```

In the example:

ELEMENT	DESCRIPTION
`\\server\Office2013`	This is the network share location that contains the Office Deployment Tool and the custom Configuration.xml file, which in this example is named **Customconfig.xml**.
`setup.exe`	This is the Office Deployment Tool.
`/packager`	This creates the Office 2013 App-V package with Volume Licensing as specified in the **Customconfig.xml** file.

ELEMENT	DESCRIPTION
`\\server\Office2013\Customconfig.xml`	This passes the configuration XML file, which in this example is named "Customconfig," that has been prepared for the packaging stage.
`\\server\share\Office2013AppV`	This specifies the location of the newly created Office App-V package.

After you run the **/packager** command, the following folders will appear in the directory where you specified the package should be saved:

- **App-V Packages**, which contains an Office 2013 App-V package and two deployment configuration files.
- **WorkingDir**

> **NOTE**
> To troubleshoot any issues, see the log files in the %temp% directory (default).

3. Verify that the Office 2013 App-V package works correctly:

   a. Publish the Office 2013 App-V package that you created globally to a test computer and verify that the Office 2013 shortcuts appear.
   b. Start a few Office 2013 applications, such as Excel or Word, to test that your package is working as expected.

Publishing the Office package for App-V

Use the following information to publish an Office package.

Methods for publishing Office App-V packages

Deploy the App-V package for Office 2013 by using the same methods you use for any other package:

- System Center Configuration Manager
- App-V Server
- Stand-alone through Windows PowerShell commands

Publishing prerequisites and requirements

PREREQUISITE OR REQUIREMENT	DETAILS
Enable Windows PowerShell scripting on the App-V clients.	To publish Office 2013 packages, you must run a script. Package scripts are disabled by default on App-V clients. To enable scripting, run the following Windows PowerShell command: `Set-AppvClientConfiguration -EnablePackageScripts 1`
Publish the Office 2013 package globally.	Extension points in the Office App-V package require installation at the computer level. When you publish at the computer level, no prerequisite actions or redistributables are needed, and the Office 2013 package globally enables its applications to work like natively installed Office, eliminating the need for administrators to customize packages.

How to publish an Office package

Run the following command to publish an Office package globally, with the bracketed value replaced by the path to the App-V package:

```
Add-AppvClientPackage <Path_to_AppV_Package> | Publish-AppvClientPackage –global
```

- From the Web Management Console on the App-V Server, you can add permissions to a group of computers instead of to a user group to enable packages to be published globally to the computers in the corresponding group.

Customizing and managing Office App-V packages

To manage your Office App-V packages, use the same operations as you would for any other package, but there are a few exceptions, as outlined in the following sections.

- Enabling Office plug-ins by using connection groups
- Disabling Office 2013 applications
- Disabling Office 2013 shortcuts
- Managing Office 2013 package upgrades
- Managing Office 2013 licensing upgrades
- Deploying Visio 2013 and Project 2013 with Office

Enabling Office plug-ins by using connection groups

Use the steps in this section to enable Office plug-ins with your Office package. To use Office plug-ins, you must use the App-V Sequencer to create a separate package that contains just the plug-ins. You can't use the Office Deployment Tool to create the plug-ins package. You then create a connection group that contains the Office package and the plug-ins package, as described in the following steps.

To enable plug-ins for Office App-V packages

1. Add a Connection Group through App-V Server, System Center Configuration Manager, or a Windows PowerShell cmdlet.

2. Sequence your plug-ins using the App-V Sequencer. Ensure that Office 2013 is installed on the computer being used to sequence the plug-in. It's a good idea to use Office 365 ProPlus (non-virtual) on the sequencing computer when you sequence Office 2013 plug-ins.

3. Create an App-V package that includes the desired plug-ins.

4. Add a Connection Group through App-V Server, System Center Configuration Manager, or a Windows PowerShell cmdlet.

5. Add the Office 2013 App-V package and the plug-ins package you sequenced to the Connection Group you created.

   > **IMPORTANT**
   >
   > The order of the packages in the Connection Group determines the order in which the package contents are merged. In your Connection group descriptor file, add the Office 2013 App-V package first, then add the plug-in App-V package.

6. Ensure that both packages are published to the target computer and that the plug-in package is published globally to match the global settings of the published Office 2013 App-V package.

7. Verify that the Deployment Configuration File of the plug-in package has the same settings that the Office

2013 App-V package has.

Since the Office 2013 App-V package is integrated with the operating system, the plug-in package settings should match. You can search the Deployment Configuration File for "COM Mode" and ensure that your plug-ins package has that value set as **Integrated** and that both **InProcessEnabled** and **OutOfProcessEnabled** match the settings of the Office 2013 App-V package you published.

8. Open the Deployment Configuration File and set the value for **Objects Enabled** to **false**.
9. If you made any changes to the Deployment Configuration file after sequencing, ensure that the plug-in package is published with the updated file.
10. Ensure that the Connection Group you created is enabled onto your desired computer. The Connection Group created will likely "pend" if the Office 2013 App-V package is in use when the Connection Group is enabled. If that happens, you'll have to reboot to successfully enable the Connection Group.
11. After you successfully publish both packages and enable the Connection Group, start the target Office 2013 application and verify that the plug-in you published and added to the Connection Group works as expected.

Disabling Office 2013 applications

You may want to disable specific applications in your Office App-V package. For instance, you can disable Access, but leave all other Office application main available. When you disable an application, the end user will no longer see the shortcut for that application. You do not have to re-sequence the application. When you change the Deployment Configuration File after the Office 2013 App-V package has been published, you will save the changes, add the Office 2013 App-V package, then republish it with the new Deployment Configuration File to apply the new settings to Office 2013 App-V Package applications.

> **NOTE**
>
> To exclude specific Office applications (for example, Access and InfoPath) when you create the App-V package with the Office Deployment Tool, use the **ExcludeApp** setting. For more information, see Reference for Click-to-Run configuration.xml file.

To disable an Office 2013 application

1. Open a Deployment Configuration File with a text editor such as **Notepad** and search for "Applications."
2. Search for the Office application you want to disable, for example, Access 2013.
3. Change the value of "Enabled" from "true" to "false."
4. Save the Deployment Configuration File.
5. Add the Office 2013 App-V Package with the new Deployment Configuration File.

```
<Application Id="[{AppVPackageRoot)]\officefl5\INFOPATH.EXE" Enabled="true">
  <VisualElements>
    <Name>InfoPath Filler 2013</Name>
    <Icon />
    <Description />
  </VisualElements>
</Application>
<Application Id="[{AppVPackageRoot}]\office15\lync.exe" Enabled="true">
  <VisualElements>
    <Name>Lync 2013</Name>
    <Icon />
    <Description />
  </VisualElements>
</Application>
<Application Id="[(AppVPackageRoot}]\office15\MSACCESS.EXE" Enabled="true">
  <VisualElements>
    <Name>Access 2013</Name>
    <Icon />
    <Description />
  </VisualElements>
</Application>
```

6. Re-add the Office 2013 App-V package, and then republish it with the new Deployment Configuration File to apply the new settings to Office 2013 App-V Package applications.

Disabling Office 2013 shortcuts

You may want to disable shortcuts for certain Office applications instead of unpublishing or removing the package. The following example shows how to disable shortcuts for Microsoft Access.

1. Open a Deployment Configuration File in Notepad and search for “Shortcuts”.

2. To disable certain shortcuts, delete or comment out the specific shortcuts you don’t want. You must keep the subsystem present and enabled. For example, in the example below, delete the Microsoft Access shortcuts, while keeping the subsystems <shortcut> </shortcut> intact to disable the Microsoft Access shortcut.

```
Shortcuts

-->
 <Shortcuts Enabled="true">
  <Extensions>
    <Extension Category="AppV.Shortcut">
      <Shortcut>
       <File>[{Common Programs}]\Microsoft Office 2013\Access 2013.lnk</File>
       <Target>[{AppvPackageRoot}])office15\MSACCESS.EXE</Target>
       <Icon>[{Windows}]\Installer\{90150000-000F-0000-0000-000000FF1CE)\accicons.exe.0.ico</Icon>
       <Arguments />
       <WorkingDirectory />
       <AppuserModelId>Microsoft.Office.MSACCESS.EXE.15</AppUserModelId>
       <AppUserModelExcludeFromShowInNewInstall>true</AppUserModelExcludeFromShowInNewInstall>
       <Description>Build a professional app quickly to manage data.</Description>
       <ShowCommand>l</ShowCommand>
       <ApplicationId>[{AppVPackageRoot}]\office15\MSACCESS.EXE</ApplicationId>
    </Shortcut>
```

3. Save the Deployment Configuration File.

4. Republish Office 2013 App-V Package with new Deployment Configuration File.

Many additional settings can be changed through modifying the Deployment Configuration for App-V packages, for example, file type associations, Virtual File System, and more. For additional information on how to use

Deployment Configuration Files to change App-V package settings, refer to the additional resources section at the end of this document.

Managing Office 2013 package upgrades

To upgrade an Office 2013 package, use the Office Deployment Tool. To upgrade a previously deployed Office 2013 package, perform the following steps.

How to upgrade a previously deployed Office 2013 package

1. Create a new Office 2013 package through the Office Deployment Tool that uses the most recent Office 2013 application software. The most recent Office 2013 bits can always be obtained through the download stage of creating an Office 2013 App-V Package. The newly created Office 2013 package will have the most recent updates and a new Version ID. All packages created using the Office Deployment Tool have the same lineage.

   > **NOTE**
   >
   > Office App-V packages have two Version IDs:
   >
   > - An Office 2013 App-V Package Version ID that is unique across all packages created using the Office Deployment Tool.
   > - A second App-V Package Version ID, x.x.x.x for example, in the AppX manifest that will only change if there is a new version of Office itself. For example, if a new Office 2013 release with upgrades is available, and a package is created through the Office Deployment Tool to incorporate these upgrades, the X.X.X.X version ID will change to reflect that the Office version itself has changed. The App-V server will use the X.X.X.X version ID to differentiate this package and recognize that it contains new upgrades to the previously published package, and as a result, publish it as an upgrade to the existing Office 2013 package.

2. Globally publish the newly created Office 2013 App-V Packages onto computers where you would like to apply the new updates. Since the new package has the same lineage of the older Office 2013 App-V Package, publishing the new package with the updates will only apply the new changes to the old package, and thus will be fast.

3. Upgrades will be applied in the same manner of any globally published App-V Packages. Because applications will probably be in use, upgrades might be delayed until the computer is rebooted.

Managing Office 2013 licensing upgrades

If a new Office 2013 App-V Package has a different license than the Office 2013 App-V Package currently deployed. For instance, the Office 2013 package deployed is a subscription based Office 2013 and the new Office 2013 package is Volume Licensing based, the following instructions must be followed to ensure smooth licensing upgrade:

How to upgrade an Office 2013 License

1. Unpublish the already deployed Office 2013 Subscription Licensing App-V package.
2. Remove the unpublished Office 2013 Subscription Licensing App-V package.
3. Restart the computer.
4. Add the new Office 2013 App-V Package Volume Licensing.
5. Publish the added Office 2013 App-V Package with Volume Licensing.

An Office 2013 App-V Package with your chosen licensing will be successfully deployed.

Deploying Visio 2013 and Project 2013 with Office

This section describes the requirements and options for deploying Visio 2013 and Project 2013 with Office.

- **To package and publish Visio 2013 and Project 2013 with Office**: Include Visio 2013 and Project 2013 in the same package with Office. If you aren't deploying Office, you can create a package that contains Visio

and/or Project.

- **To deploy Visio 2013 and Project 2013 to specific users**: Use one of the following methods:

GOAL	METHOD
Create two different packages and deploy each one to a different group of users	Create and deploy the following packages: A package that contains only Office—deploy to computers whose users need only Office. A package that contains Office, Visio, and Project—deploy to computers whose users need all three applications.
Create just one package for the whole organization, or for users who share computers	Follow these steps: 1. Create a package that contains Office, Visio, and Project. 2. Deploy the package to all users. 3. Use AppLocker to prevent specific users from using Visio and Project.

Additional resources

Additional resources for Office 2013 App-V Packages

- Office 2013 Deployment Tool for Click-to-Run
- Supported scenarios for deploying Microsoft Office as a sequenced App-V Package

Additional resources for Office 2010 App-V Packages

- Microsoft Office 2010 Sequencing Kit for Microsoft Application Virtualization 5.0
- Known issues when you create or use an App-V 5.0 Office 2010 package
- How To Sequence Microsoft Office 2010 in Microsoft Application Virtualization 5.0

Additional resources for Connection Groups

- Managing Connection Groups
- Connection groups on the App-V team blog

Additional resources for Dynamic Configuration

- About App-V Dynamic Configuration

Deploying Microsoft Office 2010 by Using App-V

5/31/2019 • 3 minutes to read • Edit Online

Applies to: Windows 10, version 1607

You can create Office 2010 packages for Microsoft Application Virtualization (App-V) using one of the following methods:

- Application Virtualization (App-V) Sequencer
- Application Virtualization (App-V) Package Accelerator

App-V support for Office 2010

The following table shows the App-V versions, methods of Office package creation, supported licensing, and supported deployments for Office 2010.

SUPPORTED ITEM	SUPPORT LEVEL
Package creation	- Sequencing - Package Accelerator - Office Deployment Kit
Supported licensing	Volume Licensing
Supported deployments	- Desktop - Personal VDI - RDS

Creating Office 2010 App-V using the sequencer

Sequencing Office 2010 is one of the main methods for creating an Office 2010 package on App-V. Microsoft has provided a detailed recipe through a Knowledge Base article. For detailed instructions about how to create an Office 2010 package on App-V, see How To Sequence Microsoft Office 2010 in Microsoft Application Virtualization 5.0.

Creating Office 2010 App-V packages using package accelerators

Office 2010 App-V packages can be created through package accelerators. Microsoft has provided package accelerators for creating Office 2010 on Windows 10, Windows 8, and Windows 7. The following pages will show you which package accelerator is best for creating Office 2010 App-V packages on your version of Windows:

- App-V 5.0 Package Accelerator for Office Professional Plus 2010 – Windows 8
- App-V 5.0 Package Accelerator for Office Professional Plus 2010 – Windows 7

For detailed instructions on how to create virtual application packages using App-V package accelerators, see How to Create a Virtual Application Package Using an App-V Package Accelerator.

Deploying the Microsoft Office package for App-V

You can deploy Office 2010 packages by using any of the following App-V deployment methods:

- System Center Configuration Manager
- App-V server
- Stand-alone through Windows PowerShell commands

Office App-V package management and customization

Office 2010 packages can be managed like any other App-V packages through known package management mechanisms. No special instructions are needed, for example, to add, publish, unpublish, or remove Office packages.

Microsoft Office integration with Windows

The following table provides a full list of supported integration points for Office 2010.

(POPULATE THE THIRD COLUMN)

EXTENSION POINT	DESCRIPTION	OFFICE 2010
Lync meeting Join Plug-in for Firefox and Chrome	User can join Lync meetings from Firefox and Chrome	
Sent to OneNote Print Driver	User can print to OneNote	Yes
OneNote Linked Notes	OneNote Linked Notes	
Send to OneNote Internet Explorer Add-In	User can send to OneNote from IE	
Firewall Exception for Lync and Outlook	Firewall Exception for Lync and Outlook	
MAPI Client	Native apps and add-ins can interact with virtual Outlook through MAPI	
SharePoint Plugin for Firefox	User can use SharePoint features in Firefox	
Mail Control Panel Applet	User gets the mail control panel applet in Outlook	Yes
Primary Interop Assemblies	Support managed add-ins	
Office Document Cache Handler	Allows Document Cache for Office applications	
Outlook Protocol Search handler	User can search in Outlook	Yes

EXTENSION POINT	DESCRIPTION	OFFICE 2010
Active X Controls: - Groove.SiteClient - PortalConnect.PersonalSite - SharePoint.openDocuments - SharePoint.ExportDatabase - SharePoint.SpreadSheetLauncher - SharePoint.StssyncHander - SharePoint.DragUploadCtl - SharePoint.DragDownloadCtl - Sharpoint.OpenXMLDocuments - Sharepoint.ClipboardCtl - WinProj.Activator - Name.NameCtrl - STSUPld.CopyCtl - CommunicatorMeetingJoinAx.JoinManager - LISTNET.Listnet - OneDrive Pro Browser Helper	Active X Control. For more information about ActiveX controls, see the ActiveX Control API Reference.	
OneDrive Pro Icon Overlays	Windows explorer shell icon overlays when users look at folders OneDrive Pro folders	

Additional resources

Office 2013 App-V Packages Additional Resources

- Supported scenarios for deploying Microsoft Office as a sequenced App-V Package

Office 2010 App-V Packages

- Microsoft Office 2010 Sequencing Kit for Microsoft Application Virtualization 5.0
- Known issues when you create or use an App-V 5.0 Office 2010 package
- How To Sequence Microsoft Office 2010 in Microsoft Application Virtualization 5.0

Connection Groups

- Managing Connection Groups
- Connection groups on the App-V team blog

Dynamic Configuration

- About App-V Dynamic Configuration

Operations for App-V

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

This section of the Microsoft Application Virtualization (App-V) Administrator's Guide includes information about the various types of App-V administration and operating tasks that are typically performed by an administrator. This section also includes step-by-step procedures to help you successfully perform those tasks.

Operations Information

- Creating and Managing App-V Virtualized Applications

  Describes how to create, modify, and convert virtualized packages.

- Administering App-V Virtual Applications by Using the Management Console

  Describes how to use the App-V Management console to perform tasks such as sequencing an application, changing a package, using a project template, and using a package accelerator.

- Managing Connection Groups

  Describes how connection groups enable virtualized applications to communicate with each other in the virtual environment; explains how to create, publish, and delete them; and describes how connection groups can help you better manage your virtualized applications.

- Deploying App-V Packages by Using Electronic Software Distribution (ESD)

  Describes how to deploy App-V packages by using an ESD.

- Using the App-V Client Management Console

  Describes how perform client configuration tasks using the client management console.

- Migrating to App-V from a Previous Version

  Provides instructions for migrating to App-V from a previous version.

- Administering App-V by Using Windows PowerShell

  Describes the set of Windows PowerShell cmdlets available for administrators performing various App-V server tasks.

Additional information

- Application Virtualization (App-V) overview
- Getting Started with App-V
- Planning for App-V
- Deploying App-V
- Troubleshooting App-V
- Technical Reference for App-V

Creating and managing App-V virtualized applications

6/10/2019 • 8 minutes to read • Edit Online

Applies to: Windows 10, version 1607

After you have properly deployed the Microsoft Application Virtualization (App-V) sequencer, you can use it to monitor and record the installation and setup process for an application to be run as a virtualized application.

For more information about configuring the App-V sequencer, sequencing best practices, and an example of creating and updating a virtual application, see the Microsoft Application Virtualization 5.0 Sequencing Guide.

> **NOTE**
> The App-V Sequencer cannot sequence applications with filenames matching "CO_<x>" where x is any numeral. Error 0x8007139F will be generated.

Sequencing an application

You can use the App-V Sequencer to perform the following tasks:

- Create virtual packages that can be deployed to computers running the App-V client.
- Upgrade existing packages. You can expand an existing package onto the computer running the sequencer and then upgrade the application to create a newer version.
- Edit configuration information associated with an existing package. For example, you can add a shortcut or modify a file type association.

  > **NOTE**
  > You must create shortcuts and save them to an available network location to allow roaming. If a shortcut is created and saved in a private location, the package must be published locally to the computer running the App-V client.

- Convert existing virtual packages.

The sequencer uses the **%TMP% \ Scratch** or **%TEMP% \ Scratch** directory and the **Temp** directory to store temporary files during sequencing. On the computer that runs the sequencer, you should configure these directories with free disk space equivalent to the estimated application installation requirements. Configuring the temp directories and the Temp directory on different hard drive partitions can help improve performance during sequencing.

When you use the sequencer to create a new virtual application, the following listed files are created. These files comprise the App-V package.

- **.msi file**. This Windows Installer (.msi) file is created by the sequencer and is used to install the virtual package on target computers.
- **Report.xml file**. In this file, the sequencer saves all issues, warnings, and errors that were discovered during sequencing. It displays the information after the package has been created. You can us this report for diagnosing and troubleshooting.

- **.appv file**. This is the virtual application file.
- **Deployment configuration file**. The deployment configuration file determines how the virtual application will be deployed to target computers.
- **User configuration file**. The user configuration file determines how the virtual application will run on target computers.

> **IMPORTANT**
> You must configure the %TMP% and %TEMP% folders that the package converter uses to be a secure location and directory. A secure location is only accessible by an administrator. Additionally, when you sequence the package, you should either save the package to a secure location or make sure that no other user is allowed to log in during the conversion and monitoring process.

The **Options** dialog box in the sequencer console contains the following tabs:

- **General**. Use this tab to enable Microsoft Updates to run during sequencing. Select **Append Package Version to Filename** to configure the sequence to add a version number to the virtualized package that is being sequenced. Select **Always trust the source of Package Accelerators** to create virtualized packages using a package accelerator without being prompted for authorization.

  > **IMPORTANT**
  > Package Accelerators created using App-V 4.6 are not supported by App-V.

- **Parse Items**. This tab displays the associated file path locations that will be parsed or tokenized into in the virtual environment. Tokens are useful for adding files using the **Package Files** tab in **Advanced Editing**.
- **Exclusion Items**. Use this tab to specify which folders and directories should not be monitored during sequencing. To add local application data that is saved in the Local App Data folder in the package, click **New** and specify the location and the associated **Mapping Type**. This option is required for some packages.

App-V supports applications that include Microsoft Windows Services. If an application includes a Windows service, the service will be included in the sequenced virtual package as long as it's installed while being monitored by the sequencer. If a virtual application creates a Windows service when it initially runs, then after installation, the application must be run while the sequencer is monitoring for the Windows Service to be included in the package. Only services running under the Local System account are supported. Services configured for AutoStart or Delayed AutoStart are started before the first virtual application in a package runs inside the package's Virtual Environment. Windows Services that are configured to be started on demand by an application are started when the virtual application inside the package starts the Service via API call.

- Automatically provision your sequencing environment using Microsoft Application Virtualization Sequencer (App-V Sequencer)
- How to Sequence a New Application with App-V
- Automatically sequence multiple apps at the same time using Microsoft Application Virtualization Sequencer (App-V Sequencer)

App-V shell extension support

App-V supports shell extensions. Shell extensions will be detected and embedded in the package during sequencing.

Shell extensions are automatically embedded in the package during the sequencing process. When the package is published, the shell extension gives users the same functionality as if the application were locally installed.

Requirements for using shell extensions

- Packages that contain embedded shell extensions must be published globally. The application requires no additional setup or configuration on the client to enable the shell extension functionality.
- The "bitness" of the application, Sequencer, and App-V client must match, or the shell extensions won't work. For example:
  - The version of the application is 64-bit.
  - The Sequencer is running on a 64-bit computer.
  - The package is being delivered to a 64-bit App-V client computer.

The following table lists the supported shell extensions:

HANDLER	DESCRIPTION
Context menu handler	Adds menu items to the context menu. It's called before the context menu is displayed.
Drag-and-drop handler	Controls the action where right-click, drag and drop, and modifies the context menu that appears.
Drop target handler	Controls the action after a data object is dragged and dropped over a drop target such as a file.
Data object handler	Controls the action after a file is copied to the clipboard or dragged and dropped over a drop target. It can provide additional clipboard formats to the drop target.
Property sheet handler	Replaces or adds pages to the property sheet dialog box of an object.
Infotip handler	Allows retrieving flags and infotip information for an item and displaying it inside a pop-up tooltip upon mouse hover.
Column handler	Allows creating and displaying custom columns in **Windows Explorer Details view**. It can be used to extend sorting and grouping.
Preview handler	Enables a preview of a file to be displayed in the Windows Explorer Preview pane.

Copy on Write (CoW) file extension support

Copy on write (CoW) file extensions allow App-V to dynamically write to specific locations contained in the virtual package while it is being used.

The following table displays the file types that can exist in a virtual package under the VFS directory, since App-V 5.1, but which cannot be updated on the computer running the App-V client. All other files and directories can be modified.

FILE TYPE					
.com	.exe	.dll	.ocx		

Modifying an existing virtual application package

You can use the sequencer to modify an existing package. The computer on which you do this should match the chip architecture of the computer you used to create the application. For example, if you initially sequenced a package using a computer running a 64-bit operating system, you should modify the package using a computer running a 64-bit operating system.

For more information, see How to Modify an Existing Virtual Application Package.

Creating a project template

An App-V project template (.appvt) file is a project template that can be used to save commonly applied, customized settings. You can then more easily use these settings for future sequencings. App-V project templates differ from App-V Application Accelerators because App-V Application Accelerators are application-specific, and App-V project templates can be applied to multiple applications. Additionally, you cannot use a project template when you use a Package Accelerator to create a virtual application package. The following general settings are saved with an App-V project template:

A template can specify and store multiple settings as follows:

- **Advanced Monitoring Options**. Enables Microsoft Update to run during monitoring. Saves allow local interaction option settings
- **General Options**. Enables the use of **Windows Installer**, **Append Package Version to Filename**.
- **Exclusion Items.** Contains the Exclusion pattern list.

In Windows 10, version 1703, running the **new-appvsequencerpackage** or **update-appvsequencepackage** cmdlets automatically captures and stores all of your customizations as an App-V project template. If you want to make changes to this package later, your customizations are automatically loaded from this template file.

> **IMPORTANT**
> If you attempt to load another template through the *TemplateFilePath* parameter while already having an auto-saved template, the customization value from the parameter will override the auto-saved template.

For more information, see How to Create and Use a Project Template.

Creating a package accelerator

> **NOTE**
> Package accelerators created using a previous version of App-V must be recreated using App-V.

You can use App-V package accelerators to automatically generate a new virtual application packages. After you have successfully created a package accelerator, you can reuse and share the package accelerator.

In some situations, to create the package accelerator, you might have to install the application locally on the computer that runs the sequencer. In such cases, you should first try to create the package accelerator with the installation media. If multiple missing files are required, you should install the application locally to the computer that runs the sequencer, and then create the package accelerator.

After you have successfully created a Package Accelerator, you can reuse and share the Package Accelerator. Creating App-V Package Accelerators is an advanced task. Package Accelerators can contain password and user-specific information. Therefore you must save Package Accelerators and the associated installation media in a secure location, and you should digitally sign the Package Accelerator after you create it so that the publisher can be verified when the App-V Package Accelerator is applied.

For more information, see the following articles:

- How to Create a Package Accelerator
- How to Create a Virtual Application Package Using an App-V Package Accelerator

Sequencer error reporting

The App-V Sequencer can detect common sequencing issues during sequencing. The **Installation Report** page at the end of the sequencing wizard displays diagnostic messages categorized into **Errors**, **Warnings**, and **Info** depending on the severity of the issue.

You can also find additional information about sequencing errors using the Windows Event Viewer.

Related topics

- Operations for App-V

Automatically provision your sequencing environment using Microsoft Application Virtualization Sequencer (App-V Sequencer)

5/31/2019 • 3 minutes to read • Edit Online

Applies to: Windows 10, version 1703

Previous versions of the App-V Sequencer have required you to manually create your sequencing environment. Windows 10, version 1703 introduces two new PowerShell cmdlets, New-AppVSequencerVM and Connect-AppvSequencerVM, which automatically create your sequencing environment for you, including provisioning your virtual machine.

Automatic VM provisioning of the sequencing environment

You have two options for provisioning an VM for auto-sequencing:

1. Using a Virtual Hard Disk (VHD)
2. Updating an existing VM

You can only choose one option.

> **NOTE**
> We have reduced the number of environmental checks performed by the App-V Sequencer, narrowing down the list of apps that need to be disabled or turned off for a clean sequencing experience. We've also suppressed antivirus and other similar app warnings.

Provision a new VM with a VHD file

Provisioning your new VM includes creating a VHD file, setting up a user account, turning on remote PowerShell scripting, and installing the App-V Sequencer.

Create a VHD file

For this process to work, you must have a base operating system available as a VHD image file, we recommend using the Convert-WindowsImage.ps1 command-line tool.

Create a VHD file with the Convert-WindowsImage command-line tool

1. Open PowerShell as an admin and run the **Convert-WindowsImage** tool, using the following commands:

   ```
   Convert-WindowsImage -SourcePath "<path_to_iso_image>" -VHDFormat "VHD" -VHDPartitionStyle "MBR"
   ```

   Where `<path_to_iso_image>` is the full path to your ISO image.

> **IMPORTANT**
> You must specify the *VHDPartitionStyle* as **MBR**. If you use the default value, **GPT**, will cause a boot failure in your VHD image.

Provision your VM with your VHD file

After you have a VHD file, you must provision your VM for auto-sequencing.

1. On the Host device, install Windows 10, version 1703 and the **Microsoft Application Virtualization (App-V) Auto Sequencer** component from the matching version of the Windows Assessment and Deployment Kit (ADK). For more info on how to install the App-V Sequencer, see Install the App-V Sequencer.

2. Make sure that Hyper-V is turned on. For more info about turning on and using Hyper-V, see Hyper-V on Windows Server 2016.

3. Open PowerShell as an admin and run the **New-AppVSequencerVM** cmdlet, using the following parameters:

   ```
   New-AppVSequencerVM -VMName "<name_of_new_vm>" -ADKPath "<path_to_adk_install_folder>" -VHDPath "
   <path_to_vhd_file>" -VMMemory "<vm_memory_size>" -VMSwitch "<name_of_network_switch>"
   ```

   This command creates a new Hyper-V VM file using the provided VHD file and also creates a "clean" checkpoint, from where all sequencing and updating will start.

Provision an existing VM for auto-sequencing

If your apps require custom prerequisites, such as Microsoft SQL Server, we recommend that you preinstall the prerequisites on your VM and then use that VM for auto-sequencing. Using these steps will establish a connection to your existing VM.

Connect to your existing VM

- Open PowerShell as an admin and run the following commands on your existing VM:

  - **Set the network category of your connection profile on the VM to *Private*:**

    ```
    Get-netconnectionprofile | set-netconnectionprofile -NetworkCategory Private
    ```

  - **Enable firewall rules for *Remote Desktop* and *Windows Remote Management*:**

    ```
    Enable-NetFirewallRule -DisplayGroup “Remote Desktop”
    Enable-NetFirewallRule -DisplayGroup “Windows Remote Management”
    ```

  - **Set the VM to receive remote commands without a confirmation prompt:**

    ```
    Enable-PSRemoting –Force
    ```

Provision an existing VM

1. On the Host device, install Windows 10, version 1703 and the **Microsoft Application Virtualization (App-V) Auto Sequencer** component from the matching version of the Windows Assessment and Deployment Kit (ADK). For more info on how to install the App-V Sequencer, see Install the App-V Sequencer.

2. Open PowerShell as an admin and run the **Connect-AppvSequencerVM** cmdlet, using the following parameters:

   ```
   Connect-AppvSequencerVM -VMName "<name_of_vm>"
   ```

   Where `<name_of_vm>` is the name of the VM as shown in the Hyper-V Manager tool.

This command connects to an existing Hyper-V VM using the provided VM name for auto-sequencing.

Review the provisioning log files

The two types of provisioning log files, located at %temp%\AutoSequencer\Logs, are:

- **New-AppVSequencerVM-<time_stamp>.txt**. Includes info about the provisioning activities, such as "Waiting for VM session", "Copying installer for Sequencer", and so on.
- **Connect-AppvSequencerVM-report-<time_stamp>.txt**. Includes info about the connections made to the VM, showing whether there were any failures.

Next steps

After provisioning your sequencing environment, you must sequence your apps, either as a group or individually. For more info about sequencing your apps, see the following articles

- Manually sequence a single new app using Microsoft Application Virtualization Sequencer (App-V Sequencer)
- Automatically sequence multiple apps at the same time using Microsoft Application Virtualization Sequencer (App-V Sequencer)
- Automatically update multiple apps at the same time using Microsoft Application Virtualization Sequencer (App-V Sequencer)

After you sequence your packages, you can automatically clean up any unpublished packages on the App-V client. To learn more, see Automatically clean up unpublished packages on the App-V client.

Related topics

- Download the **Convert-WindowsImage** tool
- Download the Windows ADK
- How to install the App-V Sequencer
- Learn about Hyper-V on Windows Server 2016

Automatically sequence multiple apps at the same time using Microsoft Application Virtualization Sequencer (App-V Sequencer)

5/31/2019 • 4 minutes to read • Edit Online

Applies to: Windows 10, version 1703

Sequencing multiple apps at the same time requires you to install and start Microsoft Application Virtualization Sequencer (App-V Sequencer), and to install the necessary apps to collect any changes made to the operating system during the installation and building of the App-V package.

In Windows 10, version 1703, running the App-V Sequencer automatically captures and stores your customizations as an App-V project template (.appvt) file. If you want to make changes to this package later, your customizations will be automatically loaded from this template file. This is applicable to all of the sequencing scenarios:

- Using the **New-BatchAppVSequencerPackages** cmdlet
- Using the App-V Sequencer interface
- Using the **New-AppVSequencerPackage** cmdlet

> **NOTE**
>
> If you're trying to update multiple apps at the same time, see Automatically update multiple apps at the same time using Microsoft Application Virtualization Sequencer (App-V Sequencer).

Sequence multiple apps with a PowerShell cmdlet

Sequencing multiple apps at the same time requires that you create a **ConfigFile** with info related to each round of sequencing. This file is then used by the cmdlet to start the VM at a "clean" checkpoint, to copy the installer from the Host device to the VM, and then to start the App-V Sequencer to monitor your specified app installations.

Create your ConfigFile for use by the PowerShell cmdlet

1. Determine the apps that need to be included in your App-V sequencing package, and then open a text editor, such as Notepad.

2. Add the following required XML info for each app:

   - `<AppName>`. The name of the app you're adding to the package.

   - `<InstallerFolder>`. The file path to the folder with the app installer.

   - `<Installer>`. The file name for the app executable. This will typically be an .exe or .msi file.

   - `<InstallerOptions>`. The command-line options required for the app installation.

   - `<TimeoutInMinutes>`. The maximum amount of time, in minutes, that the cmdlet should wait for sequencing to complete. You can enter a different value for each app, based on the size and complexity of the app itself.

   - `<Cmdlet>`. Determines whether the sequencer uses the cmdlet or the App-V Sequencer interface.

True tells the sequencer to use cmdlet-based sequencing, while **False** tells the sequencer to use the App-V Sequencer interface. You can use both the cmdlet and the interface together in the same ConfigFile, for different apps.

- `<Enabled>`. Indicates whether the app should be sequenced. **True** includes the app, while **False** ignores it. You can include as many apps as you want in the batch file, but optionally enable only a few of them.

  Example:

```
<?xml version="1.0"?>
    <Applications>
        <Application>
            <AppName>Skype for Windows</AppName>
            <InstallerFolder>D:\Install\New\SkypeforWindows</InstallerFolder>
            <Installer>SkypeSetup.exe</Installer>
            <InstallerOptions>/S</InstallerOptions>
            <TimeoutInMinutes>20</TimeoutInMinutes>
            <Cmdlet>True</Cmdlet>
            <Enabled>True</Enabled>
        </Application>
        <Application>
            <AppName>Power BI</AppName>
            <InstallerFolder>D:\Install\New\MicrosoftPowerBI</InstallerFolder>
            <Installer>PBIDesktop.msi</Installer>
            <InstallerOptions>/S</InstallerOptions>
            <TimeoutInMinutes>20</TimeoutInMinutes>
            <Cmdlet>True</Cmdlet>
            <Enabled>True</Enabled>
        </Application>
    </Applications>
</xml>
```

3. Save your completed file, using the name **ConfigFile**.

Start the App-V Sequencer interface and app installation process

Open PowerShell as an admin on the Host computer and run the following commands to start the batch sequencing:

```
New-BatchAppVSequencerPackages –ConfigFile <path_to_configfile> –VMName <name_of_vm> -OutputPath
<path_to_your_output>
```

Where `<name_of_vm>` is the name of the virtual machine (VM) with the App-V Sequencer installed, where you'll run the batch sequencing, and `<path_to_your_output>` is the full path to where the sequenced packages should be copied.

The cmdlet creates a "clean" checkpoint on the VM. Next, the cmdlet copies the first app installer listed in the ConfigFile from the Host computer to the VM, and finally a new session of the VM opens (through VMConnect) and sequencing of the app begins from the command-line. After completing sequencing and package creation for the first app on the VM, the package is copied from the VM to the Host computer, specified in the *OutputPath* parameter. The cmdlet then goes to the second app on your list, reverting the VM back to a "clean" checkpoint and running through all of the steps again, until the second app package is copied to your output folder. This process continues until all apps included in your list are done. After the last app, the VM is reverted back to a "clean" checkpoint and turned off.

Review the log files

There are 3 types of log files that occur when you sequence multiple apps at the same time:

- **New-BatchAppVSequencerPackages-<*time_stamp*>.txt**. Located in the %temp%\AutoSequencer\Logs

directory. This log contains info about the sequencing activities for each app, such as "Copying installer to VM," "Scheduling sequencing task," and so on. Additionally, if an app times out, this log contains the failure along with the checkpoint for troubleshooting the problem.

- **New-BatchAppVSequencerPackages-report-<*time_stamp*>.txt**. Located in the **OutputPath** folder you specified earlier. This log contains info about the connections made to the VM, showing if there were any failures. Additionally, it briefly includes success or failure info for all of the apps.
- **Log.txt file**. Located in the **Output Package** folder. This file contains all code included in the NewAppVSequencerPackage cmdlet, including the allowed parameters.

Related topics

- Download the Windows ADK
- How to install the App-V Sequencer
- Learn about Hyper-V on Windows Server 2016
- Automatically provision your sequencing environment using Microsoft Application Virtualization Sequencer (App-V Sequencer)
- Manually sequence a single app using Microsoft Application Virtualization Sequencer (App-V Sequencer)
- Automatically update multiple apps at the same time using Microsoft Application Virtualization Sequencer (App-V Sequencer)
- Automatically clean up unpublished packages on the App-V client

Automatically update multiple apps at the same time using Microsoft Application Virtualization Sequencer (App-V Sequencer)

5/31/2019 • 6 minutes to read • Edit Online

Applies to: Windows 10, version 1703

Updating multiple apps at the same time follows a similar process to the one used for automatically sequencing multiple apps at the same time. However, when updating, you'll also have to pass your previously created app package files to the App-V Sequencer cmdlet.

Starting with Windows 10, version 1703, running the New-BatchAppVSequencerPackages cmdlet or the App-V Sequencer interface captures and stores all of your customizations as an App-V project template. If you want to make changes to this package later, your customizations are automatically loaded from this template file.

> **NOTE**
> If you're trying to sequence multiple apps at the same time, see Automatically sequence multiple apps at the same time using Microsoft Application Virtualization Sequencer (App-V Sequencer).

Update multiple apps with a PowerShell cmdlet

Updating multiple apps at the same time requires that you create a **ConfigFile** with info related to each round of updating. This file is then used by the cmdlet to start the VM at a "clean" checkpoint, to copy the installer from the Host device to the VM, and then to start the App-V Sequencer to monitor your specified app installations.

Create your ConfigFile for use by the PowerShell cmdlet

1. Determine the apps that need to be included in your app package, and then open a text editor, such as Notepad.
2. Add the following XML info for each app:
   - `<AppName>`. The name of the app you're adding to the package.
   - `<InstallerFolder>`. The file path to the folder with the app installer.
   - `<Installer>`. The file name for the app executable. This will typically be an .exe or .msi file.
   - `<InstallerOptions>`. The command-line options required for the app installation.
   - `<Package>`. The file path to the location of your App-V packages. These packages were created when you sequenced your apps.
   - `<TimeoutInMinutes>`. The maximum amount of time, in minutes, that the cmdlet should wait for updating to complete. You can enter a different value for each app, based on the size and complexity of the app itself.
   - `<Cmdlet>`. Determines whether the sequencer uses the cmdlet or the App-V Sequencer interface. **True** tells the sequencer to use cmdlet-based updating, while **False** tells the sequencer to use the App-V Sequencer interface. You can use both the cmdlet and the interface together in the same

ConfigFile, for different apps.

- `<Enabled>`. Indicates whether the app should be sequenced. **True** includes the app, while **False** ignores it. You can include as many apps as you want in the batch file, but optionally enable only a few of them.

  Example:

```
<?xml version="1.0"?>
<Applications>
    <Application>
        <AppName>Skype for Windows Update</AppName>
        <InstallerFolder>D:\Install\Update\SkypeforWindows</InstallerFolder>
        <Installer>SkypeSetup.exe</Installer>
        <InstallerOptions>/S</InstallerOptions>
        <Package>C:\App-V_Package\Microsoft_Apps\skypeupdate.appv</Package>
        <TimeoutInMinutes>20</TimeoutInMinutes>
        <Cmdlet>true</Cmdlet>
        <Enabled>true</Enabled>
    </Application>
    <Application>
        <AppName>Microsoft Power BI Update</AppName>
        <InstallerFolder>D:\Install\Update\PowerBI</InstallerFolder>
        <Installer>PBIDesktop.msi</Installer>
        <InstallerOptions>/S</InstallerOptions>
        <Package>C:\App-V_Package\MS_Apps\powerbiupdate.appv</Package>
        <TimeoutInMinutes>20</TimeoutInMinutes>
        <Cmdlet>true</Cmdlet>
        <Enabled>true</Enabled>
    </Application>
</Applications>
```

3. Save your completed file under the name **ConfigFile**.

Start the App-V Sequencer interface and app installation process

- Open PowerShell as an admin on the Host computer and run the following commands to start the batch updating:

```
New-BatchAppVSequencerPackages –ConfigFile <path_to_configfile> –VMName <name_of_vm> -OutputPath
<path_to_your_output>
```

  Where `<name_of_vm>` is the name of the virtual machine (VM) with the App-V Sequencer installed that you'll run the batch updating on, and `<path_to_your_output>` is the full path to where the updated packages should be copied.

  The cmdlet creates a "clean" checkpoint on the VM. After making the checkpoint, the cmdlet copies the first app installer listed in the ConfigFile from the Host computer to the VM. This opens a new session of the VM (through VMConnect), allowing app updates to begin from the command-line. After completing the update and package creation for the first app on the VM, the package is copied from the VM to the Host computer, specified in the OutputPath parameter. The cmdlet then goes to the second app on your list, reverting the VM back to a "clean" checkpoint and running through all of the steps again, until the second app package is copied to your output folder. This process continues until all apps included in your list are done. After the last app, the VM is reverted back to a "clean" checkpoint and turned off.

Update multiple apps with the App-V Sequencer interface

Updating multiple apps at the same time requires that you create a **ConfigFile** to collect all of the info related to each round of updating. This file is then used by the App-V Sequencer interface after creating a "clean" checkpoint

on your VM.

Create your ConfigFile for use by the App-V Sequencer interface

1. Determine the apps that need to be updated and then open a text editor, such as Notepad.
2. Add the following XML info for each app:
   - `<AppName>`. The name of the app you're adding to the package.
   - `<InstallerFolder>`. The file path to the folder with the app installer.
   - `<Installer>`. The file name for the app executable. This will typically be an .exe or .msi file.
   - `<Package>`. The file path to the location of your App-V packages. These packages were created when you sequenced your apps.
   - `<TimeoutInMinutes>`. The maximum amount of time, in minutes, the cmdlet should wait for updating to complete. You can enter a different value for each app, based on the size and complexity of the app itself.
   - `<Cmdlet>`. Determines whether the sequencer uses the cmdlet or the App-V Sequencer interface. **True** tells the sequencer to use cmdlet-based updating, while **False** tells the sequencer to use the App-V Sequencer interface. You can use both the cmdlet and the interface together in the same ConfigFile, for different apps.
   - `<Enabled>`. Indicates whether the app should be sequenced. **True** includes the app, while **False** ignores it. You can include as many apps as you want in the batch file, but optionally enable only a few of them.

     Example:

```
<?xml version="1.0"?>
<Applications>
    <Application>
        <AppName>Skype for Windows Update</AppName>
        <InstallerFolder>D:\Install\Update\SkypeforWindows</InstallerFolder>
        <Installer>SkypeSetup.exe</Installer>
        <InstallerOptions>/S</InstallerOptions>
        <Package>C:\App-V_Package\Microsoft_Apps\skypeupdate.appv</Package>
        <TimeoutInMinutes>20</TimeoutInMinutes>
        <Cmdlet>false</Cmdlet>
        <Enabled>true</Enabled>
    </Application>
    <Application>
        <AppName>Microsoft Power BI Update</AppName>
        <InstallerFolder>D:\Install\Update\PowerBI</InstallerFolder>
        <Installer>PBIDesktop.msi</Installer>
        <InstallerOptions>/S</InstallerOptions>
        <Package>C:\App-V_Package\MS_Apps\powerbiupdate.appv</Package>
        <TimeoutInMinutes>20</TimeoutInMinutes>
        <Cmdlet>false</Cmdlet>
        <Enabled>true</Enabled>
    </Application>
</Applications>
```

Start the App-V Sequencer interface and app installation process

- Open PowerShell as an admin on the Host computer and run the following commands to start the batch updating:

```
New-BatchAppVSequencerPackages –ConfigFile <path_to_configfile> –VMName <name_of_vm> -OutputPath
<path_to_your_output>
```

Where `<name_of_vm>` is the name of the virtual machine (VM) with the App-V Sequencer installed, where you'll run the batch updating, and `<path_to_your_output>` is the full path to where the updated packages should be copied.

This cmdlet creates a "clean" checkpoint on the VM. After making the checkpoint, the cmdlet copies the first app installer listed in the ConfigFile from the Host computer to the VM. This opens a new session of the VM (through VMConnect) and app updating begins from the command-line. After completing updating and package creation for the first app on the VM, the package is copied from the VM to the Host computer specified in the *OutputPath* parameter. The cmdlet then goes to the second app on your list, reverting the VM back to a "clean" checkpoint and running through all of the steps again, until the second app package is copied to your output folder. This process continues until all apps included in your list are done. After the last app, the VM is reverted to a "clean" checkpoint and turned off.

Review the log files

There are three types of log files that occur when you sequence multiple apps at the same time:

- **New-BatchAppVSequencerPackages-<*time_stamp*>.txt**. Located in the %temp%\AutoSequencer\Logs directory. This log contains info about updating activities for each app, such as "Copying installer to VM," "Scheduling updating task," and so on. Additionally, if an app times out, this log contains the failure along with the checkpoint for troubleshooting the problem.
- **New-BatchAppVSequencerPackages-report-<*time_stamp*>.txt**. Located in the **OutputPath** folder you specified earlier. This log contains info about the connections made to the VM, showing if there were any failures. Additionally, it briefly includes success or failure info for all of the apps.
- **Log.txt file**. Located in the **Output Package** folder. This file contains all code included in the **NewAppVSequencerPackage** cmdlet, including the allowed parameters.

Related topics

- Download the Windows ADK
- How to install the App-V Sequencer
- Learn about Hyper-V on Windows Server 2016
- Automatically provision your sequencing environment using Microsoft Application Virtualization Sequencer (App-V Sequencer)
- Manually sequence a single app using Microsoft Application Virtualization Sequencer (App-V Sequencer)
- Automatically sequence multiple apps at the same time using Microsoft Application Virtualization Sequencer (App-V Sequencer)
- Automatically cleanup unpublished packages on the App-V client

Manually sequence a new app using the Microsoft Application Virtualization Sequencer (App-V Sequencer)

6/10/2019 • 14 minutes to read • Edit Online

Applies to: Windows 10, version 1607 and later

In Windows 10, version 1607, the App-V Sequencer is included with the Windows ADK. For more info on how to install the App-V Sequencer, see Install the App-V Sequencer.

Before you start sequencing

1. Determine the type of virtualized application package you want to create:

APPLICATION TYPE	DESCRIPTION
Standard	Creates a package that contains an application or a suite of applications. This is the preferred option for most application types.
Add-on or plug-in	Creates a package that extends the functionality of a standard application, for example, a plug-in for Microsoft Excel. Additionally, you can use plug-ins for natively installed applications, or for another package that is linked by using connection groups.
Middleware	Creates a package that is required by a standard application, for example, Java. Middleware packages are used for linking to other packages through connection groups.

2. Copy all required installation files to the computer that is running the sequencer.
3. Make a backup image of your virtual environment before sequencing an application, and then revert to that image each time after you finish sequencing an application.
4. Review the following items:
   - If an application installer changes the security access to a new or existing file or directory, those changes are not captured in the package.
   - If short paths have been disabled for the virtualized package's target volume, you must also sequence the package to a volume that was created and still has short-paths disabled. It cannot be the system volume.

> **NOTE**
>
> The App-V Sequencer cannot sequence applications with filenames matching "CO_<*x*>" where *x* is any numeral. Error 0x8007139F will be generated.

Sequence a new standard application

1. On the computer that runs the sequencer, select **All Programs**, and then select **Microsoft Application Virtualization**, and then select **Microsoft Application Virtualization Sequencer**.

2. In the sequencer, select **Create a New Virtual Application Package**. Select **Create Package (default)**, and then select **Next**.

3. On the **Prepare Computer** page, review the issues that could cause the package creation to fail or could cause the package to contain unnecessary data. You should resolve all potential issues before you continue. After making any corrections, select **Refresh** to display the updated information. After you have resolved all potential issues, select **Next**.

   > **IMPORTANT**
   > If you are required to disable virus scanning software, you should first scan the computer that runs the sequencer in order to ensure that no unwanted or malicious files could be added to the package.

4. On the **Type of Application** page, select the **Standard Application (default)** check box, and then select **Next**.

5. On the **Select Installer** page, select **Browse** and specify the installation file for the application.

   > **NOTE**
   > If the specified application installer modifies security access to a file or directory, existing or new, the associated changes will not be captured into the package.

   If the application does not have an associated installer file and you plan to run all installation steps manually, select the **Perform a Custom Installation** check box, and then select **Next**.

6. On the **Package Name** page, specify a name for the package. Use a name that helps identify the purpose and version of the application that will be added to the package. The package name is displayed in the App-V Management Console. Once you're done, select **Next**.

7. On the **Installation** page, when the sequencer and application installer are ready, you can install the application so that the sequencer can monitor the installation process.

   > **IMPORTANT**
   > You should always install applications to a secure location and make sure no other users are logged on to the computer running the sequencer during monitoring.

   Use the application's installation process to perform the installation. If additional installation files must be run as part of the installation, select **Run** to locate and run the additional installation files. When you are finished with the installation, select **I am finished installing**, then select **Next**.

8. On the **Installation** page, wait while the sequencer configures the virtualized application package.

9. On the **Configure Software** page, optionally run the programs contained in the package. This step allows you to complete any necessary license or configuration tasks before you deploy and run the package on target computers. To run all the programs at one time, select at least one program, and then select **Run All**. To run specific programs, select the program or programs, and then select **Run Selected**. Complete the required configuration tasks and then close the applications. You may need to wait several minutes for all programs to run.

> **NOTE**
> To run first-use tasks for any application that is not available in the list, open the application. The associated information will be captured during this step.

Select **Next**.

10. On the **Installation Report** page, you can review information about the virtualized application package you have just sequenced. In **Additional Information**, double-click an event to obtain more detailed information. To proceed, select **Next**.

11. The **Customize** page is displayed. If you've finished installing and configuring the virtual application, select **Stop now** and skip to step 14 of this procedure. To perform either of the following customizations, select **Customize**.

    - Prepare the virtual package for streaming. Streaming improves the experience when the virtual application package is run on target computers.
    - Specify the operating systems that can run this package.

    Once you're ready, select**Next**.

12. On the **Streaming** page, run each program so that it can be optimized and run more efficiently on target computers. It can take several minutes for all the applications to run. After all applications have run, close each of the applications, and then select **Next**.

> **NOTE**

```
>If you do not open any applications during this step, the default streaming method is on-demand streaming
delivery. This means applications will be downloaded bit by bit until it can be opened. After that, depending
on how the background loading is configured, it will load the rest of the application.
```

13. On the **Target OS** page, specify the operating systems that can run this package. To allow all supported operating systems in your environment to run this package, select **Allow this package to run on any operating system**. To configure this package to run only on specific operating systems, select **Allow this package to run only on the following operating systems** and select the operating systems that can run this package. After that, select **Next**.

    > **IMPORTANT**
    > Make sure that the operating systems you specify here are supported by the application you are sequencing.

14. The **Create Package** page is displayed. To modify the package without saving it, select **Continue to modify package without saving using the package editor**. This option opens the package in the sequencer console so that you can modify the package before saving it. Select **Next**.

    To save the package immediately, select **Save the package now** (default). Add optional **Comments** to be associated with the package. Comments are useful for identifying the program version and other information about the package.

    > **IMPORTANT**
    > The system does not support non-printable characters in **Comments** and **Descriptions**.

The default **Save Location** is also displayed on this page. To change the default location, select **Browse** and specify the new location. After that, select **Create**.

15. The **Completion** page is displayed. Review the information in the **Virtual Application Package Report** pane as needed, then select **Close**. This information is also available in the **Report.xml** file located in the directory where the package was created.

    Your package should now be available in the sequencer.

    > **IMPORTANT**
    > After you have successfully created a virtual application package, you can't run the virtual application package on the computer that is running the sequencer.

Sequence an add-on or plug-in application

> **NOTE**
> Before performing the following procedure, install the parent application locally on the computer that is running the sequencer. Or if you have the parent application virtualized, you can follow the steps in the add-on or plug-in workflow to unpack the parent application on the computer. For example, if you are sequencing a plug-in for Microsoft Excel, install Microsoft Excel locally on the computer that's running the sequencer. You should also install the parent application in the same directory where the application is installed on target computers. If the plug-in or add-on is going to be used with an existing virtual application package, install the application on the same virtual application drive that was used when you created the parent virtual application package.

1. On the computer that runs the sequencer, first, select **All Programs**, then select **Microsoft Application Virtualization**, and then select **Microsoft Application Virtualization Sequencer**.

2. In the sequencer, select **Create a New Virtual Application Package**, select **Create Package (default)**, and then select **Next**.

3. On the **Prepare Computer** page, review the issues that might cause the package creation to fail or could cause the package to contain unnecessary data. You should resolve all potential issues before you continue. After making any corrections, select **Refresh** to display the updated information. After you have resolved all potential issues, select **Next**.

    > **IMPORTANT**
    > If you are required to disable virus scanning software, you should first scan the computer that runs the sequencer in order to ensure that no unwanted or malicious files could be added to the package.

4. On the **Type of Application** page, select **Add-on or Plug-in**, and then select **Next**.

5. On the **Select Installer** page, select **Browse** and specify the installation file for the add-on or plug-in. If the add-on or plug-in does not have an associated installer file and you plan to run all installation steps manually, select the **Select this option to perform a custom installation** check box, then select **Next**.

6. On the **Install Primary** page, ensure that the primary application is installed on the computer that runs the sequencer. Alternatively, you can expand an existing package that has been saved locally on the computer that runs the sequencer. To do this, select **Expand Package**, and then select the package. After you have expanded or installed the parent program, select **I have installed the primary parent program**.

7. Select **Next**.

8. On the **Package Name** page, type a name that will be associated with the package. Use a name that helps identify the purpose and version of the application that will be added to the package. The package name will be displayed in the App-V Management Console.

9. Select **Next**.

10. On the **Installation** page, when the sequencer and application installer are ready you can proceed to install the plug-in or add-in application so the sequencer can monitor the installation process. Use the application's installation process to perform the installation. If additional installation files must be run as part of the installation, select **Run** and locate and run the additional installation files. When you are finished with the installation, select **I am finished installing**, and then select **Next**.

11. On the **Installation Report** page, you can review information about the virtual application package that you just sequenced. For a more detailed explanation about the information displayed in **Additional Information**, double-click the event. After you have reviewed the information, select **Next**.

12. The **Customize** page is displayed. If you are finished installing and configuring the virtual application, select **Stop now** and skip to step 12 of this procedure. To perform either of the following customizations, select **Customize**.

    - Optimize how the package will run across a slow or unreliable network.
    - Specify the operating systems that can run this package.

    When you're finished, select **Next**.

13. On the **Streaming** page, run each program so that it can be optimized and run more efficiently on target computers. Streaming improves the experience when the virtual application package is run on target computers on high-latency networks. It can take several minutes for all applications to run. After all applications have run, close each application. You can also configure the package to be required to be fully downloaded before opening by selecting the **Force applications to be downloaded** check-box. Select **Next**.

NOTE

```
>If necessary, you can stop an application from loading during this step. In the **Application Launch**
dialog box, select **Stop** and select one of the check boxes: **Stop all applications** or **Stop this
application only**.
```

14. On the **Target OS** page, specify the operating systems that can run this package. To allow all supported operating systems in your environment to run this package, select the **Allow this package to run on any operating system** check box. To configure this package to run only on specific operating systems, select the **Allow this package to run only on the following operating systems** check box, and then select the operating systems that can run this package. Select **Next**.

15. The **Create Package** page is displayed. To modify the package without saving it, select **Continue to modify package without saving using the package editor** check box. This option opens the package in the sequencer console so that you can modify the package before it is saved. Select **Next**.

    To save the package immediately, select **Save the package now**. Optionally, add a **Description** for the package. Descriptions are useful for identifying the version and other important information about the package.

> **IMPORTANT**
> The system does not support non-printable characters in Comments and Descriptions.

The default **Save Location** is also displayed on this page. To change the default location, select **Browse** and specify the new location. Select **Create**.

Sequence a middleware application

1. On the computer that runs the sequencer, select **All Programs**, then select **Microsoft Application Virtualization**, and then select **Microsoft Application Virtualization Sequencer**.
2. In the sequencer, select **Create a New Virtual Application Package**, select **Create Package (default)**, and then select **Next**.
3. On the **Prepare Computer** page, review the issues that could cause the package creation to fail or could cause the package to contain unnecessary data. You should resolve all potential issues before you continue. After making any corrections, select **Refresh** to display the updated information. After you have resolved all potential issues, select **Next**.

> **IMPORTANT**
> If you are required to disable virus scanning software, you should first scan the computer that runs the App-V Sequencer in order to ensure that no unwanted or malicious files can be added to the package.

4. On the **Type of Application** page, select **Middleware**, and then select **Next**.
5. On the **Select Installer** page, select **Browse** and specify the installation file for the application. If the application does not have an associated installer file and you plan to run all installation steps manually, select the **Select this option to perform a custom installation** check box, then select **Next**.
6. On the **Package Name** page, type a name that will be associated with the package. Use a name that helps identify the purpose and version of the application that will be added to the package. The package name is displayed in the App-V Management Console.
7. Select **Next**.
8. On the **Installation** page, when the sequencer and middleware application installer are ready you can proceed to install the application so that the sequencer can monitor the installation process. Use the application's installation process to perform the installation. If additional installation files must be run as part of the installation, select **Run**, to locate and run the additional installation files. When you are finished with the installation, select the **I am finished installing** check box, and then select **Next**.
9. On the **Installation** page, wait while the sequencer configures the virtual application package.
10. On the **Installation Report** page, you can review information about the virtual application package that you have just sequenced. In **Additional Information**, double-click an event to obtain more detailed information. To proceed, select **Next**.
11. On the **Target OS** page, specify the operating systems that can run this package. To enable all supported operating systems in your environment to run this package, select the **Allow this package to run on any operating system** check box. To configure this package to run only on specific operating systems, select the **Allow this package to run only on the following operating systems** check box and select the operating systems that can run this package. Once you're done, select **Next**.
12. On the **Create Package** page is displayed. To modify the package without saving it, select **Continue to modify package without saving using the package editor**. This option opens the package in the

sequencer console so that you can modify the package before it is saved. Select **Next**.

To save the package immediately, select **Save the package now**. Optionally, add a **Description** to be associated with the package. Descriptions are useful for identifying the program version and other information about the package.

> **IMPORTANT**
>
> The system does not support non-printable characters in comments and descriptions.

The default **Save Location** is also displayed on this page. To change the default location, select **Browse** and specify the new location. Select **Create**.

13. The **Completion** page is displayed. Review the information in the **Virtual Application Package Report** pane as needed, then select **Close**. This information is also available in the **Report.xml** file that is located in the directory specified in step 11 of this procedure.

    The package is now available in the sequencer. To edit the package properties, select **Edit [Package Name]**.

    > **IMPORTANT**
    >
    > After you have successfully created a virtual application package, you can't run the virtual application package on the computer that is running the sequencer.

Related topics

- Install the App-V Sequencer
- Operations for App-V

How to Modify an Existing Virtual Application Package

5/31/2019 • 8 minutes to read • Edit Online

Applies to

- Windows 10, version 1607

This topic explains how to:

- Update an application in an existing virtual application package
- Modify the properties associated with an existing virtual application package
- Add a new application to an existing virtual application package

Before you update a package:

- Ensure that you've installed the Microsoft Application Virtualization (App-V) Sequencer, which is required for modifying a virtual application package. To install the App-V Sequencer, see How to Install the Sequencer.
- Save the .appv file in a secure location and always trust the source before trying to open the package for editing.
- The Managing Authority section is erroneously removed from the deployment configuration file when you update a package. Before starting the update, copy the Managing Authority section from the existing deployment configuration file, and then paste the copied section into the new configuration file after the conversion is complete.
- If you click **Modify an Existing Virtual Application Package** in the Sequencer in order to edit a package, but then make no changes and close the package, the streaming behavior of the package is changed. The primary feature block is removed from the StreamMap.xml file, and any files that were listed in the publishing feature block are removed. Users who receive the edited package experience that package as if it were stream-faulted, regardless of how the original package was configured.

Update an application in an existing virtual application package

1. On the computer that runs the sequencer, click **All Programs**, point to **Microsoft Application Virtualization**, and then click **Microsoft Application Virtualization Sequencer**.
2. In the App-V Sequencer, click **Modify an Existing Virtual Application Package** > **Next**.
3. On the **Select Task** page, click **Update Application in Existing Package** > **Next**.
4. On the **Select Package** page, click **Browse** to locate the virtual application package that contains the application to update, and then click **Next**.
5. On the **Prepare Computer** page, review the issues that could cause the application update to fail or cause the updated application to contain unnecessary data. Resolve all potential issues before you continue. After making any corrections and resolving all potential issues, click **Refresh** > **Next**.

   Important If you are required to disable virus scanning software, first scan the computer that runs the sequencer to ensure that no unwanted or malicious files are added to the package.

6. On the **Select Installer** page, click **Browse** and specify the update installation file for the application. If the update does not have an associated installer file, and if you plan to run all installation steps manually, select the **Select this option to perform a custom installation** check box, and then click **Next**.

7. On the **Installation** page, when the sequencer and application installer are ready you can proceed to install the application update so the sequencer can monitor the installation process. If additional installation files must be run as part of the installation, click **Run**, and then locate and run the additional installation files. When you are finished with the installation, select **I am finished installing**. Click **Next**.

   > **Note** The sequencer monitors all changes and installations that occur on the computer that runs the sequencer. This includes any changes and installations that are performed outside of the sequencing wizard.

8. On the **Installation Report** page, you can review information about the updated virtual application. In **Additional Information**, double-click the event to obtain more detailed information. To proceed, click **Next**.

9. On the **Streaming** page, run each program so that it can be optimized and run more efficiently on target computers. It can take several minutes for all of the applications to run. After all applications have run, close each of the applications, and then click **Next**.

   > **Note** You can stop an application from loading during this step. In the **Application Launch** dialog box, click **Stop**, and then select either **Stop all applications** or **Stop this application only**.

10. On the **Create Package** page, to modify the package without saving it, select the check box for **Continue to modify package without saving using the package editor**. When you select this option, the package opens in the App-V Sequencer console, where you can modify the package before it is saved. Click **Next**.

    To save the package immediately, select the default **Save the package now**. Add optional **Comments** to associate with the package. Comments are useful to identify the application version and provide other information about the package. The default **Save Location** is also displayed. To change the default location, click **Browse** and specify the new location. Click **Create**.

11. On the **Completion** page, click **Close** to close the wizard. The package is now available in the sequencer.

Modify the properties associated with an existing virtual application package

1. On the computer that runs the sequencer, click **All Programs**, point to **Microsoft Application Virtualization**, and then click **Microsoft Application Virtualization Sequencer**.

2. In the App-V Sequencer, click **Modify an Existing Virtual Application Package** > **Next**.

3. On the **Select Task** page, click **Edit Package** > **Next**.

4. On the **Select Package** page, click **Browse** to locate the virtual application package that contains the application properties to modify, and then click **Edit**.

5. In the App-V Sequencer console, perform any of the following tasks as needed:

   - Import and export the manifest file.
   - Enable or disable Browser Helper Objects.
   - Import or export a VFS file.
   - Import a directory into the virtual file system.

- Import and export virtual registry keys.
- View package properties.
- View associated package files.
- Edit registry settings.
- Review additional package settings (except operating system file properties).
- Set virtualized registry key state (override or merge).
- Set virtualized folder state.
- Add or edit shortcuts and file type associations.

> **Note** To edit shortcuts or file type associations, you must first open the package for upgrade to add a new application, and then proceed to the final editing page.

6. When you finish changing the package properties, click **File** > **Save** to save the package.

Add a new application to an existing virtual application package

1. On the computer that runs the sequencer, click **All Programs**, point to **Microsoft Application Virtualization**, and then click **Microsoft Application Virtualization Sequencer**.
2. In the App-V Sequencer, click **Modify an Existing Virtual Application Package** > **Next**.
3. On the **Select Task** page, click **Add New Application** > **Next**.
4. On the **Select Package** page, click **Browse** to locate the virtual application package to which you will add the application, and then click **Next**.
5. On the **Prepare Computer** page, review the issues that could cause the package creation to fail or cause the revised package to contain unnecessary data. Resolve all potential issues before you continue. After making any corrections and resolving all potential issues, click **Refresh** > **Next**.

> **Important** If you are required to disable virus scanning software, first scan the computer that runs the sequencer to ensure that no unwanted or malicious files can be added to the package.

6. On the **Select Installer** page, click **Browse** and specify the installation file for the application. If the application does not have an associated installer file and you plan to run all installation steps manually, select the **Select this option to perform a custom installation** check box, and then click **Next**.
7. On the **Installation** page, when the sequencer and application installer are ready, install the application so that the sequencer can monitor the installation process. If additional installation files must be run as part of the installation, click **Run**, and locate and run the additional installation files. When you finish the installation, select **I am finished installing** > **Next**. In the **Browse for Folder** dialog box, specify the primary directory where the application will be installed. Ensure that this is a new location so that you don't overwrite the existing version of the virtual application package.

> **Note** The sequencer monitors all changes and installations that occur on the computer that runs the sequencer. This includes any changes and installations that are performed outside of the sequencing wizard.

8. On the **Configure Software** page, optionally run the programs contained in the package. This step completes any associated license or configuration tasks that are required to run the application before you

deploy and run the package on target computers. To run all the programs at the same time, select at least one program, and then click **Run All**. To run specific programs, select the program or programs you want to run, and then click **Run Selected**. Complete the required configuration tasks and then close the applications. It can take several minutes for all programs to run. Click **Next**.

9. On the **Installation Report** page, you can review information about the updated virtual application. In **Additional Information**, double-click the event to obtain more detailed information, and then click **Next** to open the **Customize** page.

10. If you are finished installing and configuring the virtual application, select **Stop now** and skip to step 13 of this procedure. If you want to perform the following described customization, click **Customize**.

    If you are customizing, prepare the virtual package for streaming, and then click **Next**. Streaming improves the experience when the virtual application package is run on target computers.

11. On the **Streaming** page, run each program so that it can be optimized and run more efficiently on target computers. It can take several minutes for all the applications to run. After all applications have run, close each of the applications, and then click **Next**.

    > **Note** You can stop an application from loading during this step. In the **Application Launch** dialog box, click **Stop** and then select either **Stop all applications** or **Stop this application only**.

12. On the **Create Package** page, to modify the package without saving it, select the **Continue to modify package without saving using the package editor** check box. Selecting this option opens the package in the App-V Sequencer console, where you can modify the package before saving it. Click **Next**.

    To save the package immediately, select the default **Save the package now**. Add optional **Comments** to associate with the package. Comments are useful for providing application versions and other information about the package. The default **Save Location** is also displayed. To change the default location, click **Browse** and specify the new location. The uncompressed package size is displayed. Click **Create**.

13. On the **Completion** page, click **Close**. The package is now available in the sequencer.

For App-V issues, use the App-V TechNet Forum.

Related topics

Operations for App-V

Create and apply an App-V project template to a sequenced App-V package

6/10/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

You can use an App-V Project Template (.appvt) file to save commonly applied settings associated with an existing virtual application package. You can then apply these settings whenever you create new virtual application packages in your environment, streamlining the package creation process. App-V Project Templates differ from App-V Package Accelerators because App-V Package Accelerators are application-specific, while App-V Project Templates can be applied to multiple applications. To learn more about package accelerators, see How to create a package accelerator.

> **IMPORTANT**
>
> In Windows 10, version 1703, running the **New-AppvSequencerPackage** or the **Update-AppvSequencerPackage** cmdlets will automatically capture and store your customizations as an App-V Project Template. If you want to make changes to this package later, you can automatically load your customizations from this template file. If you have an auto-saved template and you attempt to load another template through the *TemplateFilePath* parameter, the customization value from the parameter will override the auto-saved template.

Create a project template

You must first create and save a project template, including a virtual app package with settings to be used by the template.

1. On the device running the App-V Sequencer, select **Start**, select **All Programs**, select **Microsoft Application Virtualization**, and then select **Microsoft Application Virtualization Sequencer**.

   > **NOTE**
   >
   > If the virtual app package is currently open in the App-V Sequencer console, skip to Step 3 of this procedure.

2. On the **File** menu, select **Open**, select **Edit Package**, browse for the virtual app package that includes the settings you want to save with the App-V Project Template, and then select **Edit** to change any of the settings or info included in the file.

3. On the **File** menu, select **Save As Template**, review the settings associated with the new template, select **OK**, name your new template, and then select **Save**.

   The new App-V Project Template is saved in the folder you specified.

Apply a project template

After creating the template, you can apply it to all of your new virtual app packages, automatically including all of the settings.

> **IMPORTANT**
>
> Virtual app packages don't support using both a project template and a package accelerator at the same time.

1. On the device running the App-V Sequencer, select **Start** > **All Programs** > **Microsoft Application Virtualization** > **Microsoft Application Virtualization Sequencer**.
2. On the **File** menu, select **New From Template**, browse to your newly created project template and select **Open**.
3. Create your new virtual app package. The settings saved with your template are automatically applied.

Related topics

- Download the Windows ADK
- How to install the App-V Sequencer
- Learn about Hyper-V on Windows Server 2016
- Automatically sequence multiple apps at the same time using Microsoft Application Virtualization Sequencer (App-V Sequencer)
- Automatically update multiple apps at the same time using Microsoft Application Virtualization Sequencer (App-V Sequencer)
- Manually sequence a new app using Microsoft Application Virtualization Sequencer (App-V Sequencer)

How to create a package accelerator

6/10/2019 • 3 minutes to read • Edit Online

Applies to: Windows 10, version 1607

App-V Package Accelerators automatically generate new virtual application packages.

> **NOTE**
>
> You can use Windows PowerShell to create a package accelerator. For more information, see How to create a package accelerator by using Windows PowerShell.

Use the following procedure to create a package accelerator.

> **IMPORTANT**
>
> - Because package accelerators can contain password and user-specific information, you should save package accelerators and the associated installation media in a secure location, and you should also digitally sign the package accelerator after creating it so that you can verify the publisher when applying the App-V Package Accelerator.
> - Before you begin creating a package accelerator, do the following:
> - Copy the virtual application package that you will use to create the package accelerator locally to the computer running the sequencer.
> - Copy all required installation files associated with the virtual application package to the computer running the sequencer.
> - The App-V Sequencer does not grant any license rights to the software application you are using to create the package accelerator. You must abide by all end user license terms for the application you are using. It is your responsibility to make sure the software application's license terms allow you to create a package accelerator with the App-V sequencer.

Create a package accelerator

1. To start the App-V sequencer on the computer running the sequencer, select **Start** > **All Programs** > **Microsoft Application Virtualization** > **Microsoft Application Virtualization Sequencer**.

2. To start the App-V **Create Package Accelerator** wizard, in the App-V sequencer console, select **Tools** > **Create Accelerator**.

3. On the **Select Package** page, select **Browse** to specify an existing virtual application package to use to create the package accelerator, then locate the existing virtual application package (it will appear as an .appv file).

   > **TIP**
   >
   > Copy the files associated with the virtual application package you plan to use locally to the computer running the Sequencer.

   Select **Next**.

4. Go to the **Installation Files** page and select **Browse**, then select the directory that contains the installation files to specify the folder containing the original virtual package's installation files.

> **TIP**
> Copy the folder that contains the required installation files to the computer running the Sequencer.

5. If the application is already installed on the computer running the sequencer, then select **Files installed on local system** to specify the installation file. To use this option, the application must already be installed in the default installation location.

6. On the **Gathering Information** page, review the files that you couldn't find in the location specified by the **Installation Files** page. If the files displayed are not required, select **Remove these files**, then select **Next**. If the files are required, select **Previous** and copy the required files to the directory specified on the **Installation Files** page.

> **NOTE**
> You must either remove the unrequired files or select **Previous** and locate the required files to advance to the next page of this wizard.

7. On the **Select Files** page, carefully review the detected files. Clear any file the package accelerator doesn't need to run successfully and select only the files that the application requires. When you're done, select **Next**.

8. Confirm that the **Verify Applications** page displays all installation files required to build the package. The package accelerator requires all installation files displayed in the **Applications** pane in order to create the package.

   If you need to add additional Installer files, select **Add**. To remove unnecessary installation files, select the **Installer file**, then select **Delete**. To edit the properties associated with an installer, select **Edit**. The package accelerator requires the installation files specified in this step to create a new virtual application package. After you have confirmed the information displayed, select **Next**.

9. On the **Select Guidance** page, select **Browse** to specify the file that will provide the package accelerator with application instructions. For example, this file can contain information about how the computer running the Sequencer should be configured, application prerequisite information for target computers, and general notes. You should provide all required information for successful package accelerator application. The file you select must be in rich text (.rtf) or text file (.txt) format. After specifying the file, select **Next**.

10. On the **Create Package Accelerator** page, select **Browse** and select the directory where you want to save the package accelerator.

11. On the **Completion** page, select **Close**.

> **IMPORTANT**
> You should always digitally sign the package accelerator to ensure that it is secure and can be verified by a publisher during application.

Related topics

- Operations for App-V
- How to create a virtual application package using an App-V Package Accelerator

How to create a virtual application package using an App-V Package Accelerator

6/10/2019 • 4 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Use the following procedure to create a virtual application package with the App-V Package Accelerator.

> **IMPORTANT**
> The App-V Sequencer does not grant any license rights to the software application that you use to create the package accelerator. You must abide by all end user license terms for the application that you use. It is your responsibility to make sure that the software application's license terms allow you to create a package accelerator with the App-V Sequencer.

Create a virtual application package with an App-V Package Accelerator

1. Make sure you've copied the required package accelerator locally to the computer running the App-V Sequencer. Also make sure to copy all required installation files for the package to a local folder on the computer running the Sequencer. This is the folder that you have to specify in step 6 of this procedure.

2. To start the App-V Sequencer on the computer that runs the Sequencer, go to **Start** > **All Programs** > **Microsoft Application Virtualization** > **Microsoft Application Virtualization Sequencer**.

3. Select **Create a New Virtual Application Package**. To create the package, select the **Create Package using a Package Accelerator** check box, then select **Next**.

4. To specify the package accelerator that will be used to create the new virtual application package, select **Browse** on the **Select Package Accelerator** page. Select **Next**.

   > **IMPORTANT**
   > If the publisher of the package accelerator cannot be verified and does not contain a valid digital signature, then before you select **Run**, you must confirm that you trust the source of the package accelerator. Confirm your choice in the **Security Warning** dialog box.

5. On the **Guidance** page, review the publishing guidance information that is displayed in the information pane. This information was added when the package accelerator was created and it contains guidance about how to create and publish the package. To export the guidance information to a text (.txt) file, select **Export** and specify the location where the file should be saved, and then select **Next**.

6. On the **Select Installation Files** page, select **Make New Folder** to create a local folder that contains all required installation files for the package, and specify where the folder should be saved. You must also specify a name to be assigned to the folder. You must then copy all required installation files to the location that you specified. If the folder that contains the installation files already exists on the computer that runs the Sequencer, select **Browse** to select the folder.

   Alternatively, if you have already copied the installation files to a directory on this computer, select **Make New Folder**, browse to the folder that contains the installation files, then select **Next**.

> **NOTE**
>
> You can specify the following types of supported installation files:
>
> - Windows Installer files (**.msi**)
> - Cabinet files (.cab)
> - Compressed files with a .zip file name extension
> - The actual application files The following file types are not supported: **.msp** and **.exe** files. If you specify an **.exe** file, you must extract the installation files manually.

7. If the package accelerator requires you to install an application before you apply the package accelerator and you have already installed the required application, select **I have installed all applications**, then select **Next** on the **Local Installation** page.

8. On the **Package Name** page, specify a name that will be associated with the package. The name you choose will identify the package in the App-V Management Console. Select **Next**.

9. On the **Create Package** page, provide comments that will be associated with the package. The comments should contain identifying information about the package that you are creating. To confirm the location where the package is created, review the information displayed in **Save Location**. To compress the package, select **Compress Package**. Select the **Compress Package** check box if the package will be streamed across the network or the package size exceeds 4 GB.

10. To create the package, select **Create**. After the package is created, select **Next**.

11. On the **Configure Software** page, to enable the Sequencer to configure the applications contained within the package, select **Configure Software**. **Configure Software** will let you configure any associated tasks required to run the application on the target computers. For example, you can configure any associated license agreements.

    The following items can be configured using the Sequencer as part of this step:

    - **Load Package** loads files associated with the package. It can take several seconds to an hour to decode the package.
    - **Run Each Program** optionally runs programs contained within the package. This step can help you complete associated license or configuration tasks that must be completed before deploying and running the package on target computers. To run all the programs at once, select at least one program, and then select **Run All**. To run specific programs, select the program or programs that you want to run, and then select **Run Selected**. Complete the required configuration tasks, then close the applications. It can take several minutes for all programs to run. Select **Next**.
    - **Save Package** saves the package.
    - **Primary Feature Block** optimizes the package for streaming by rebuilding the primary feature block.

    If you don't want to configure the applications, select **Skip this step**, then select **Next**.

12. On the **Completion** page, after you review the information that is displayed in the **Virtual Application Package Report** pane, select **Close**.

    The package is now available in the Sequencer. To edit the package properties, select **Edit [Package Name]**. For more information about how to modify a package, see How to modify an existing virtual application package.

Related topics

- Operations for App-V

Administering App-V Virtual Applications by using the Management Console

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Use the Microsoft Application Virtualization (App-V) management server to manage packages, connection groups, and package access in your environment. The server publishes application icons, shortcuts, and file type associations to authorized computers running the App-V client. One or more management servers typically share a common data store for configuration and package information.

The management server uses Active Directory Domain Services (AD DS) groups to manage user authorization and has SQL Server installed to manage the database and data store.

Because the management servers stream applications to end users on demand, these servers are ideally suited for system configurations that have reliable, high-bandwidth LANs. The management server consists of the following components:

- The **management server** manages packages and connection groups.
- The **publishing server** deploys packages to computers running the App-V Client.
- The **management database** manages the package access publishes the server's synchronization with the management server.

Management Console tasks

Here are some articles that can show you how to perform the most common tasks that the App-V Management Console is used for:

- How to connect to the Management Console
- How to add or upgrade packages by using the Management Console
- How to configure access to packages by using the Management Console
- How to publish a package by using the Management Console
- How to delete a package in the Management Console
- How to add or remove an administrator by using the Management Console
- How to register and unregister a publishing server by using the Management Console
- How to create a custom configuration file by using the App-V Management Console
- How to transfer access and configurations to another version of a package by using the Management Console
- How to customize virtual application extensions for a specific AD group by using the Management Console
- How to view and configure applications and default virtual application extensions by using the Management Console

The main elements of the App-V Management Console are:

MANAGEMENT CONSOLE TAB	DESCRIPTION
Packages tab	Use the **Packages** tab to add or upgrade packages.

MANAGEMENT CONSOLE TAB	DESCRIPTION
Connection Groups tab	Use the **Connection Groups** tab to manage connection groups.
Servers tab	Use the **Servers** tab to register a new server.
Administrators tab	Use the **Administrators** tab to register, add, or remove administrators in your App-V environment.

> **IMPORTANT**
>
> The browser you're using to open the Web Management Console must have JavaScript enabled.

Other resources for this App-V deployment

- Application Virtualization (App-V) overview
- Operations for App-V

About App-V dynamic configuration

6/10/2019 • 15 minutes to read • Edit Online

Applies to: Windows 10, version 1607

You can use dynamic configuration to customize an App-V package for a user. This article will tell you how to create or edit an existing dynamic configuration file.

When you edit the Dynamic Configuration file, it customizes how an App-V package will run for a user or group. This makes package customization more convenient by removing the need to resequence packages using the desired settings and provides a way to keep package content and custom settings independent.

Advanced: dynamic configuration

Virtual application packages contain a manifest that provides all the core information for the package. This information includes the defaults for the package settings and determines settings in the most basic form (with no additional customization). If you want to adjust these defaults for a particular user or group, you can create and edit the following files:

- User Configuration file
- Deployment Configuration file

These .xml files specify package settings let you customize packages without directly affecting the packages. When a package is created, the sequencer automatically generates default deployment and user configuration .xml files using the package manifest data. These automatically generated configuration files reflect the package's default settings that were configured during sequencing. If you apply these configuration files to a package in the form generated by the sequencer, the packages will have the same default settings that came from their manifest. This provides you with a package-specific template to get started if any of the defaults must be changed.

> **NOTE**
> The following information can only be used to modify sequencer generated configuration files to customize packages to meet specific user or group requirements.

Dynamic Configuration file contents

All of the additions, deletions, and updates in the configuration files need to be made in relation to the default values specified by the package's manifest information. The following list represents the relationship between these files in how they'll be read, from most to least precedence:

- User Configuration .xml file
- Deployment Configuration .xml file
- Package Manifest

The first item represents what will be read last. Therefore, its content takes precedence. All packages inherently contain and provide default settings from the Package Manifest, but it also has the least precedence. If you apply a Deployment Configuration .xml file with customized settings, it will override the Package Manifest's defaults. If you apply a User Configuration .xml file with customized settings prior to that, it will override both the deployment configuration and the Package Manifest's defaults.

There are two types of configuration files:

- **User Configuration file (UserConfig)**: Allows you to specify or modify custom settings for a package. These settings will be applied for a specific user when the package is deployed to a computer running the App-V client.
- **Deployment Configuration file (DeploymentConfig)**: Allows you to specify or modify the default settings for a package. These settings will be applied for all users when a package is deployed to a computer running the App-V client.

You can use the UserConfig file to customize the settings for a package for a specific set of users on a computer or make changes that will be applied to local user locations such as HKCU. You can use the DeploymentConfig file to modify the default settings of a package for all users on a machine or make changes that will be applied to global locations such as HKEY_LOCAL_MACHINE and the All Users folder.

The UserConfig file provides configuration settings that you can apply to a single user without affecting any other users on a client:

- Extensions that will be integrated into the native system per user: shortcuts, File-Type associations, URL Protocols, AppPaths, Software Clients, and COM.
- Virtual Subsystems: Application Objects, Environment variables, Registry modifications, Services, and Fonts.
- Scripts (user context only).

The DeploymentConfig file provides configuration settings in two sections, one relative to the machine context and one relative to the user context providing the same capabilities listed in the preceding UserConfig list:

- All UserConfig settings from the preceding section in this topic
- Extensions that can only be applied globally for all users
- Virtual Subsystems that can be configured for global machine locations, such as the registry
- Product Source URL
- Scripts (Machine context only)
- Controls to terminate child processes

File structure

The structure of the App-V Dynamic Configuration file is explained in the following section.

Dynamic User Configuration file

The following is an example of a Dynamic User Configuration file's header:

```
<?xml version="1.0" encoding="utf-8"?>
<UserConfiguration PackageId="1f8488bf-2257-46b4-b27f-09c9dbaae707" DisplayName="Reserved"
xmlns="http://schemas.microsoft.com/appv/2010/userconfiguration">
```

The **PackageId** is the same value that exists in the Manifest file.

Dynamic User Configuration file body

The Dynamic User Configuration file's body can include all app extension points defined in the Manifest file, as well as information to configure virtual applications. There are four subsections allowed in the body:

Applications: All app-extensions contained in the Manifest file within a package are assigned with an Application ID, which is also defined in the manifest file. This allows you to enable or disable all the extensions for a given application within a package. The **Application ID** must exist in the Manifest file or it will be ignored.

```
    <UserConfiguration PackageId="1f8488bf-2257-46b4-b27f-09c9dbaae707" DisplayName="Reserved"
xmlns="http://schemas.microsoft.com/appv/2010/userconfiguration">
    <Applications>
    <!-- No new application can be defined in policy. AppV Client will ignore any application ID that is not
also in the Manifest file -->
    <Application Id="{a56fa627-c35f-4a01-9e79-7d36aed8225a}" Enabled="false">
    </Application>
    </Applications>
    …
    </UserConfiguration>
```

Subsystems: AppExtensions and other subsystems are arranged as subnodes under `<Subsystems>`, as shown in the following example.

```
    <UserConfiguration **PackageId**="1f8488bf-2257-46b4-b27f-09c9dbaae707" DisplayName="Reserved"
xmlns="http://schemas.microsoft.com/appv/2010/userconfiguration">
    <Subsystems>
    ..
    </Subsystems>
    ..
    </UserConfiguration>
```

Each subsystem can be enabled/disabled using the **Enabled** attribute. The following sections describe the various subsystems and usage samples.

Dynamic User Configuration file extensions

Extension Subsystems control extensions. These subsystems are Shortcuts, File-Type associations, URL Protocols, AppPaths, Software Clients, and COM.

Extension Subsystems can be enabled and disabled independently of the content. Therefore, if Shortcuts are enabled, the client will use the shortcuts contained within the manifest by default. Each Extension Subsystem can contain an `<Extensions>` node. If this child element is present, the client will ignore the content in the Manifest file for that subsystem and only use the content in the configuration file.

Examples of the shortcuts subsystem

Example 1

Content will be ignored if the user defined the following in either the dynamic or deployment config file:

```
<Shortcuts  Enabled="true">
            <Extensions
             ...
            </Extensions>
</Shortcuts>
```

Example 2

Content in the manifest will be integrated during publishing if the user defined only the following:

```
`<Shortcuts  Enabled="true"/>`
```

Example 3

All shortcuts in the manifest will be ignored and no shortcuts will be integrated if the user defines the following:

```
                            <Shortcuts  Enabled="true">
                                                <Extensions/>
                                      </Shortcuts>
```

Supported Extension Subsystems

Shortcuts: This controls shortcuts that will be integrated into the local system. The following example has two shortcuts:

```
    <Subsystems>
    <Shortcuts Enabled="true">
      <Extensions>
        <Extension Category="AppV.Shortcut">
          <Shortcut>
            <File>\[{Common Programs}\]\\Microsoft Contoso\\Microsoft ContosoApp Filler 2010.lnk</File>
            <Target>\[{PackageRoot}\]\\Contoso\\ContosoApp.EXE</Target>
            <Icon>\[{Windows}\]\\Installer\\{90140000-0011-0000-0000-0000000FF1CE}\\inficon.exe</Icon>
            <Arguments />
            <WorkingDirectory />
            <AppUserModelId>ContosoApp.Filler.3</AppUserModelId>
            <Description>Fill out dynamic forms to gather and reuse information throughout the organization
using Microsoft ContosoApp.</Description>
            <Hotkey>0</Hotkey>
            <ShowCommand>1</ShowCommand>
            <ApplicationId>\[{PackageRoot}\]\\Contoso\\ContosoApp.EXE</ApplicationId>
          </Shortcut>
      </Extension>
      <Extension Category="AppV.Shortcut">
        <Shortcut>
          <File>\[{AppData}\]\\Microsoft\\Contoso\\Recent\\Templates.LNK</File>
          <Target>\[{AppData}\]\\Microsoft\\Templates</Target>
          <Icon />
          <Arguments />
          <WorkingDirectory />
          <AppUserModelId />
          <Description />
          <Hotkey>0</Hotkey>
          <ShowCommand>1</ShowCommand>
          <!-- Note the ApplicationId is optional -->
        </Shortcut>
      </Extension>
     </Extensions>
    </Shortcuts>
```

File Type Associations: Associates file types with programs to open by default as well as setup the context menu. (MIME types can also be set up with this subsystem). The following is an example of a FileType association:

```
    <FileTypeAssociations Enabled="true">
    <Extensions>
      <Extension Category="AppV.FileTypeAssociation">
        <FileTypeAssociation>
          <FileExtension MimeAssociation="true">
          <Name>.docm</Name>
          <ProgId>contosowordpad.DocumentMacroEnabled.12</ProgId>
          <PerceivedType>document</PerceivedType>
          <ContentType>application/vnd.ms-contosowordpad.document.macroEnabled.12</ContentType>
          <OpenWithList>
            <ApplicationName>wincontosowordpad.exe</ApplicationName>
          </OpenWithList>
         <OpenWithProgIds>
            <ProgId>contosowordpad.8</ProgId>
          </OpenWithProgIds>
          <ShellNew>
            <Command />
            <DataBinary />
            <DataText />
            <FileName />
            <NullFile>true</NullFile>
            <ItemName />
            <IconPath />
            <MenuText />
            <Handler />
          </ShellNew>
        </FileExtension>
        <ProgId>
           <Name>contosowordpad.DocumentMacroEnabled.12</Name>
            <DefaultIcon>\[{Windows}\]\\Installer\\{90140000-0011-0000-0000-
0000000FF1CE}\\contosowordpadicon.exe,15</DefaultIcon>
            <Description>Blah Blah Blah</Description>
            <FriendlyTypeName>\[{FOLDERID\_ProgramFilesX86}\]\\Microsoft Contoso
14\\res.dll,9182</FriendlyTypeName>
            <InfoTip>\[{FOLDERID\_ProgramFilesX86}\]\\Microsoft Contoso 14\\res.dll,1424</InfoTip>
            <EditFlags>0</EditFlags>
            <ShellCommands>
              <DefaultCommand>Open</DefaultCommand>
              <ShellCommand>
                 <ApplicationId>{e56fa627-c35f-4a01-9e79-7d36aed8225a}</ApplicationId>
                 <Name>Edit</Name>
                 <FriendlyName>&Edit</FriendlyName>
                 <CommandLine>"\[{PackageRoot}\]\\Contoso\\WINcontosowordpad.EXE" /vu "%1"</CommandLine>
              </ShellCommand>
              </ShellCommand>
                <ApplicationId>{e56fa627-c35f-4a01-9e79-7d36aed8225a}</ApplicationId>
                <Name>Open</Name>
                <FriendlyName>&Open</FriendlyName>
                <CommandLine>"\[{PackageRoot}\]\\Contoso\\WINcontosowordpad.EXE" /n "%1"</CommandLine>
                <DropTargetClassId />
                <DdeExec>
                  <Application>mscontosowordpad</Application>
                  <Topic>ShellSystem</Topic>
                  <IfExec>\[SHELLNOOP\]</IfExec>
                  <DdeCommand>\[SetForeground\]\[ShellNewDatabase "%1"\]</DdeCommand>
                </DdeExec>
              </ShellCommand>
            </ShellCommands>
          </ProgId>
         </FileTypeAssociation>
       </Extension>
      </Extensions>
      </FileTypeAssociations>
```

URL Protocols: This controls the URL Protocols integrated into the local registry of the client machine. The following example illustrates the “mailto:” protocol.

```
    <URLProtocols Enabled="true">
    <Extensions>
    <Extension Category="AppV.URLProtocol">
    <URLProtocol>
      <Name>mailto</Name>
      <ApplicationURLProtocol>
      <DefaultIcon>\[{ProgramFilesX86}\]\\Microsoft Contoso\\Contoso\\contosomail.EXE,-9403</DefaultIcon>
      <EditFlags>2</EditFlags>
      <Description />
      <AppUserModelId />
      <FriendlyTypeName />
      <InfoTip />
    <SourceFilter />
      <ShellFolder />
      <WebNavigableCLSID />
      <ExplorerFlags>2</ExplorerFlags>
      <CLSID />
      <ShellCommands>
      <DefaultCommand>open</DefaultCommand>
      <ShellCommand>
      <ApplicationId>\[{ProgramFilesX86}\]\\Microsoft Contoso\\Contoso\\contosomail.EXE</ApplicationId>
      <Name>open</Name>
      <CommandLine>\[{ProgramFilesX86}\\Microsoft Contoso\\Contoso\\contosomail.EXE" -c OEP.Note /m "%1"
</CommandLine>
      <DropTargetClassId />
      <FriendlyName />
      <Extended>0</Extended>
      <LegacyDisable>0</LegacyDisable>
      <SuppressionPolicy>2</SuppressionPolicy>
       <DdeExec>
      <NoActivateHandler />
      <Application>contosomail</Application>
      <Topic>ShellSystem</Topic>
      <IfExec>\[SHELLNOOP\]</IfExec>
      <DdeCommand>\[SetForeground\]\[ShellNewDatabase "%1"\]</DdeCommand>
      </DdeExec>
      </ShellCommand>
      </ShellCommands>
      </ApplicationURLProtocol>
      </URLProtocol>
      </Extension>
      </Extension>
      </URLProtocols>
```

Software Clients: Allows the app to register as an email client, news reader, or media player and makes the app visible in the Set Program Access and Computer Defaults UI. In most cases, you only need to enable and disable it. There's also a control that lets you enable or disable the email client only in case you want all the other clients to remain as they are.

```
    <SoftwareClients Enabled="true">
      <ClientConfiguration EmailEnabled="false" />
    </SoftwareClients>
```

AppPaths: If an application, such as contoso.exe, is registered with an apppath name of “myapp”, this subsystem lets you open the app by entering “myapp” into the run menu.

```
    <AppPaths Enabled="true">
    <Extensions>
    <Extension Category="AppV.AppPath">
    <AppPath>
      <ApplicationId>\[{ProgramFilesX86}\]\\Microsoft Contoso\\Contoso\\contosomail.EXE</ApplicationId>
      <Name>contosomail.exe</Name>
      <ApplicationPath>\[{ProgramFilesX86}\]\\Microsoft Contoso\\Contoso\\contosomail.EXE</ApplicationPath>
      <PATHEnvironmentVariablePrefix />
      <CanAcceptUrl>false</CanAcceptUrl>
      <SaveUrl />
    </AppPath>
    </Extension>
    </Extensions>
    </AppPaths>
```

COM: Allows an Application to register Local COM servers. Mode can be Integration, Isolated or Off. When Isol.

```
    <COM Mode="Isolated"/>
```

Other settings for Dynamic User Configuration file

In addition to Extensions, the following other subsystems can be enabled/disabled and edited.

Virtual Kernel Objects

```
    <Objects Enabled="false" />
```xml

**Virtual Registry**: use this if you want to set a registry in the Virtual Registry within HKCU.

```xml
    <Registry Enabled="true">
    <Include>
    <Key Path="\\REGISTRY\\USER\\\[{AppVCurrentUserSID}\]\\Software\\ABC">
    <Value Type="REG\_SZ" Name="Bar" Data="NewValue" />
     </Key>
      <Key Path="\\REGISTRY\\USER\\\[{AppVCurrentUserSID}\]\\Software\\EmptyKey" />
     </Include>
    <Delete>
      </Registry>
```

Virtual File System

```
          <FileSystem Enabled="true" />
```

Virtual Fonts

```
          <Fonts Enabled="false" />
```

Virtual Environment Variables

```
    <EnvironmentVariables Enabled="true">
    <Include>
           <Variable Name="UserPath" Value="%path%;%UserProfile%" />
           <Variable Name="UserLib" Value="%UserProfile%\\ABC" />
           </Include>
          <Delete>
           <Variable Name="lib" />
            </Delete>
            </EnvironmentVariables>
```

Virtual services

```
        <Services Enabled="false" />
```

UserScripts

Scripts can be used to set up or alter the virtual environment and execute scripts on deployment or removal, before an application executes, or they can clean up the environment after the application terminates. Please refer to a sample User Configuration file output by the sequencer to see a sample script. See the Scripts section for more information about the various triggers you can use to set up scripts.

Dynamic Deployment Configuration file

Dynamic Deployment Configuration file header

The header of a Deployment Configuration file should look something like this:

```
<?xml version="1.0" encoding="utf-8"?><DeploymentConfiguration PackageId="1f8488bf-2257-46b4-b27f-
09c9dbaae707" DisplayName="Reserved" xmlns="http://schemas.microsoft.com/appv/2010/deploymentconfiguration">
```

The **PackageId** is the same value as the one that exists in the Manifest file.

Dynamic Deployment Configuration file body

The body of the deployment configuration file includes two sections:

- The User Configuration section allows the same content as the User Configuration file described in the previous section. When the package is published to a user, any appextensions configuration settings in this section will override corresponding settings in the Manifest within the package unless a user configuration file is also provided. If a UserConfig file is also provided, it will be used instead of the User settings in the deployment configuration file. If the package is published globally, then only the contents of the deployment configuration file will be used in combination with the manifest.
- The Machine Configuration section contains information that can only be configured for an entire machine, not for a specific user on the machine. For example, HKEY_LOCAL_MACHINE registry keys in the VFS.

```
<DeploymentConfiguration PackageId="1f8488bf-2257-46b4-b27f-09c9dbaae707" DisplayName="Reserved"
xmlns="http://schemas.microsoft.com/appv/2010/deploymentconfiguration">
<UserConfiguration>
..
</UserConfiguration>
<MachineConfiguration>
..
</MachineConfiguration>
..
</MachineConfiguration>
</DeploymentConfiguration>
```

User Configuration: see Dynamic User Configuration for more information about this section.

Machine Configuration: The Machine Configuration section of the Deployment Configuration File configures information that can only be set for an entire machine, not a specific user on the computer, like the HKEY_LOCAL_MACHINE registry keys in the Virtual Registry. This element can have the following four subsections.

Subsystems

AppExtensions and other subsystems are arranged as subnodes under `<Subsystems>`:

```
<MachineConfiguration>
  <Subsystems>
  ..
  </Subsystems>
..
</MachineConfiguration>
```

The following section describes the various subsystems and usage samples.

Extensions

Some subsystems (Extension Subsystems) control extensions that can only apply to all users. The subsystem is application capabilities. Because this can only apply to all users, the package must be published globally in order for this type of extension to be integrated into the local system. The rules for User Configuration extension controls and settings also apply to the ones in Machine Configuration.

Application Capabilities

Used by default programs in the Windows OS interface, the Application Capabilities extension allows an application to register itself as capable of opening certain file extensions, as a contender for the Start menu's internet browser slot, and as capable of opening certain Windows MIME types. This extension also makes the virtual application visible in the Set Default Programs UI.

```
<ApplicationCapabilities Enabled="true">
  <Extensions>
   <Extension Category="AppV.ApplicationCapabilities">
    <ApplicationCapabilities>
     <ApplicationId>\[{PackageRoot}\]\\LitView\\LitViewBrowser.exe</ApplicationId>
     <Reference>
      <Name>LitView Browser</Name>
      <Path>SOFTWARE\\LitView\\Browser\\Capabilities</Path>
     </Reference>
   <CapabilityGroup>
    <Capabilities>
     <Name>@\[{ProgramFilesX86}\]\\LitView\\LitViewBrowser.exe,-12345</Name>
     <Description>@\[{ProgramFilesX86}\]\\LitView\\LitViewBrowser.exe,-12346</Description>
     <Hidden>0</Hidden>
     <EMailSoftwareClient>Lit View E-Mail Client</EMailSoftwareClient>
     <FileAssociationList>
      <FileAssociation Extension=".htm" ProgID="LitViewHTML" />
      <FileAssociation Extension=".html" ProgID="LitViewHTML" />
      <FileAssociation Extension=".shtml" ProgID="LitViewHTML" />
     </FileAssociationList>
     <MIMEAssociationList>
      <MIMEAssociation Type="audio/mp3" ProgID="LitViewHTML" />
      <MIMEAssociation Type="audio/mpeg" ProgID="LitViewHTML" />
     </MIMEAssociationList>
    <URLAssociationList>
      <URLAssociation Scheme="http" ProgID="LitViewHTML.URL.http" />
     </URLAssociationList>
     </Capabilities>
  </CapabilityGroup>
   </ApplicationCapabilities>
  </Extension>
</Extensions>
</ApplicationCapabilities>
```

Other settings for Dynamic Deployment Configuration file

You can edit other subsystems in addition to extensions:

- Machine-wide Virtual Registry: use this when you want to set a registry key in the virtual registry within HKEY_Local_Machine.

```
<Registry>
<Include>
  <Key Path="\\REGISTRY\\Machine\\Software\\ABC">
    <Value Type="REG\_SZ" Name="Bar" Data="Baz" />
   </Key>
  <Key Path="\\REGISTRY\\Machine\\Software\\EmptyKey" />
 </Include>
<Delete>
</Registry>
```

- Machine-wide Virtual Kernel Objects

```
<Objects>
<NotIsolate>
   <Object Name="testObject" />
 </NotIsolate>
</Objects>
```

- ProductSourceURLOptOut: Indicates whether the URL for the package can be modified globally through PackageSourceRoot to support branch office scenarios. It's set to False by default. Changes to the value take effect on the next launch.

```
<MachineConfiguration>
  ..
  <ProductSourceURLOptOut Enabled="true" />
  ..
</MachineConfiguration>
```

- MachineScripts: The package can be configured to execute scripts upon deployment, publishing, or removal. To see an example script, please see a sample deployment configuration file generated by the sequencer. The following section provides more information about the various triggers you can use to set up scripts.

- TerminateChildProcess: you can use this to specify that an application executable's child processes will be terminated when the application.exe process is terminated.

```
<MachineConfiguration>
  ..
  <TerminateChildProcesses>
    <Application Path="\[{PackageRoot}\]\\Contoso\\ContosoApp.EXE" />
    <Application Path="\[{PackageRoot}\]\\LitView\\LitViewBrowser.exe" />
    <Application Path="\[{ProgramFilesX86}\]\\Microsoft Contoso\\Contoso\\contosomail.EXE" />
  </TerminateChildProcesses>
  ..
</MachineConfiguration>
```

Scripts

The following table describes the various script events and the context under which they can be run.

SCRIPT EXECUTION TIME	CAN BE SPECIFIED IN DEPLOYMENT CONFIGURATION	CAN BE SPECIFIED IN USER CONFIGURATION	CAN RUN IN THE PACKAGE'S VIRTUAL ENVIRONMENT	CAN BE RUN IN THE CONTEXT OF A SPECIFIC APPLICATION	RUNS IN SYSTEM/USER CONTEXT: (DEPLOYMENT CONFIGURATION, USER CONFIGURATION)
AddPackage	X				(SYSTEM, N/A)
PublishPackage	X	X			(SYSTEM, User)
UnpublishPackage	X	X			(SYSTEM, User)
RemovePackage	X				(SYSTEM, N/A)
StartProcess	X	X	X	X	(User, User)
ExitProcess	X	X		X	(User, User)
StartVirtualEnvironment	X	X	X		(User, User)
TerminateVirtualEnvironment	X	X			(User, User)

Using multiple scripts on a single event trigger

App-V supports the use of multiple scripts on a single event trigger for App-V packages, including packages that you convert from App-V 4.6 to App-V for Windows 10. To enable the use of multiple scripts, App-V uses a script launcher application, named ScriptRunner.exe, which is included in the App-V client.

How to use multiple scripts on a single event trigger

For each script that you want to run, pass that script as an argument to the ScriptRunner.exe application. The application will run each script separately, along with the arguments that you specify for each script. Use only one script (ScriptRunner.exe) per trigger.

> **NOTE**
>
> We recommended you first run the multi-script line from a command prompt to make sure all arguments are built correctly before adding them to the deployment configuration file.

Example script and parameter descriptions

Using the following example file and table, modify the deployment or user configuration file to add the scripts that you want to run.

```
<MachineScripts>
 <AddPackage>
   <Path>ScriptRunner.exe</Path>
   <Arguments>
   -appvscript script1.exe arg1 arg2 –appvscriptrunnerparameters –wait –timeout=10
   -appvscript script2.vbs arg1 arg2
   -appvscript script3.bat arg1 arg2 –appvscriptrunnerparameters –wait –timeout=30 –rollbackonerror
   </Arguments>
   <Wait timeout=”40” RollbackOnError=”true”/>
 </AddPackage>
</MachineScripts>
```

PARAMETER IN THE EXAMPLE FILE	DESCRIPTION
`<AddPackage>`	Name of the event trigger you're running a script for, such as when adding or publishing a package.
`ScriptRunner.exe`	The script launcher application included in the App-V client. Although ScriptRunner.exe is included in the App-V client, the App-V client's location must be in %path% or ScriptRunner won't run. `ScriptRunner.exe` is typically located in the C:\Program Files\Microsoft Application Virtualization\Client folder.

`-appvscript script1.exe arg1 arg2 –appvscriptrunnerparameters –wait –timeout=10`
 	
`-appvscript script2.vbs arg1 arg2`
 	
`-appvscript script3.bat arg1 arg2 –appvscriptrunnerparameters –wait –timeout=30 -rollbackonerror`
	`-appvscript` —token that represents the actual script you want to run.
	`script1.exe` —name of the script you want to run.
	`arg1 arg2` —arguments for the script you want to run.
	`-appvscriptrunnerparameters` —token that represents the execution options for script1.exe.
	`-wait` —token that tells ScriptRunner to wait for execution of script1.exe to finish before proceeding to the next script.
	`-timeout=x` —token that informs ScriptRunner to stop running the current script after *x* number of seconds. All other specified scripts will still run.
	`-rollbackonerror` —token that tells ScriptRunner to stop running all scripts that haven't yet run and roll back an error to the App-V client.

`<Wait timeout=”40” RollbackOnError=”true”/>`	Waits for overall completion of ScriptRunner.exe. Set the timeout value for the overall runner to be greater than or equal to the sum of the timeout values on the individual scripts. If any individual script reported an error and rollbackonerror was set to True, then ScriptRunner should report the error to App-V client.

ScriptRunner will run any script whose file type is associated with an application installed on the computer. If the associated application is missing, or the script’s file type isn't associated with any of the computer's applications, the script won't run.

Create a Dynamic Configuration file using an App-V Manifest file

You can create the Dynamic Configuration file using one of three methods: manually, using the App-V Management Console, or by sequencing a package, which will generate a package with two sample files.

For more information about how to create the file using the App-V Management Console, see How to create a

Custom Configuration file by using the App-V Management Console.

To create the file manually, you can combine the components listed in the previous sections into a single file. However, we recommend you use files generated by the sequencer instead of manually created ones.

Related topics

- How to Apply the Deployment Configuration File by Using Windows PowerShell
- How to Apply the User Configuration File by Using Windows PowerShell
- Operations for App-V

How to connect to the Management Console

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Use the following procedure to connect to the App-V Management Console.

Connect to the App-V Management Console

1. Open your web browser and enter the address for the App-V Management server. For example, **https://<*management server name*>:<*management service port number*>/console.html**.

2. To view different sections of the console, select your desired section in the navigation pane.

Related topics

- Operations for App-V

How to add or upgrade packages by using the Management Console

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

You can use the following procedure to add or upgrade a package to the App-V Management Console. To upgrade a package that already exists in the Management Console, use the following steps and import the upgraded package using the same package **Name**.

Add a package to the Management Console

1. Select the **Packages** tab in the navigation pane of the Management Console display.

   The console displays the list of packages that have been added to the server along with status information about each package. When a package is selected, detailed information about the package is displayed in the **PACKAGES** pane.

   Select the **Ungrouped** drop-down list box and specify how the packages are to be displayed in the console. You can also click the associated column header to sort the packages.

2. Select **Add or Upgrade Packages** to specify which package you want to add.

3. Enter the full path to the package that you want to add. Use the UNC or HTTP path format, for example **\\servername\sharename\foldername\packagename.appv** or **http://server.1234/file.appv**, and then select **Add**.

   > **IMPORTANT**
   >
   > You must select a package with the **.appv** file name extension.

4. The page displays the status message **Adding <Packagename>**. Select **IMPORT STATUS** to check the status of a package that you have imported.

   Select **OK** to add the package and close the **Add Package** page. If there was an error during the import, select **Detail** on the **Package Import** page for more information. The newly added package is now available in the **PACKAGES** pane.

5. Select **Close** to close the **Add or Upgrade Packages** page.

Related topics

- Operations for App-V

How to configure access to packages by using the Management Console

6/10/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Before you deploy an App-V virtualized package, you must configure the Active Directory Domain Services (AD DS) security groups that will be allowed to access and run the applications. The security groups may contain computers or users. Entitling a package to a computer group publishes the package globally to all computers in the group.

Use the following procedure to configure access to virtualized packages.

Grant access to an App-V package

1. Find the package you want to configure:
   a. Open the App-V Management console.
   b. Right-click the package to be configured, then select **Edit active directory access** to display the **AD Access** page. Alternatively, select the package and select **Edit** in the **AD Access** pane.
2. Provision a security group for the package:
   a. Go to the **Find valid Active Directory names and grant access** page.
   b. Using the format **mydomain** \ **groupname**, enter the name or part of the name of an Active Directory group object, then select **Check**.

   > **NOTE**
   > Ensure that you provide an associated domain name for the group that you are searching for.

3. Grant access to the package by first selecting the desired group, then selecting **Grant Access**. The newly added group is displayed in the **AD entities with access** pane.
4. Select **Close** to accept the default configuration settings and close the AD Access page.

   To customize configurations for a specific group, select the **Assigned configurations** drop-down menu, then select **Custom**. To make changes to your custom configurations, select **Edit**. After you grant access, select **Close**.

Remove access to an App-V package

1. Find the package you want to configure:
   a. Open the App-V Management console.
   b. To display the **AD Access** page, right-click the package to be configured, then select **Edit active directory access**. Alternatively, select the package, then select **Edit** in the **AD Access** pane.
2. Select the group you want to remove, then select **Delete**.

3. Select **Close**.

Related topics

- Operations for App-V

How to publish a package by using the Management console

6/10/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Use the following procedure to publish an App-V package. Once you publish a package, computers running the App-V client can access and run the applications in that package.

> **NOTE**
> The ability to enable only administrators to publish or unpublish packages (described below) is supported starting in App-V 5.0 SP3.

Publish an App-V package

1. In the App-V Management console. Select or right-click the name of the package to be published. Select **Publish**.
2. Review the **Status** column to verify that the package has been published and is now available. If the package is available, the status **published** is displayed.

   If the package is not published successfully, the status **unpublished** is displayed, along with error text that explains why the package is not available.

Enable only administrators to publish or unpublish packages

1. Navigate to the following Group Policy Object node:

   Computer Configuration > **Administrative Templates** > **System** > **App-V** > **Publishing**.
2. Enable the **Require publish as administrator** Group Policy setting.

   To instead use Windows PowerShell to set this item, see Understanding pending packages: UserPending and GlobalPending.

Related topics

- Operations for App-V
- How to configure access to packages by using the Management console

How to delete a package in the Management Console

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Use the following procedure to delete an App-V package.

Delete a package in the Management Console

1. To view the package you want to delete, open the App-V Management Console and select **Packages**. Select the package to be removed.
2. Select or right-click the package, then select **Delete** to remove the package.

Related topics

- Operations for App-V

How to add or remove an administrator by using the Management Console

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Use the following procedures to add or remove an administrator on the Microsoft Application Virtualization (App-V) server.

Add an administrator using the Management Console

1. Open the Microsoft Application Virtualization (App-V) Management Console and select **Administrators** in the navigation pane. The navigation pane will display a list of Access Directory (AD) users and groups that currently have administrative access to the Microsoft Application Virtualization (App-V) server.
2. To add a new administrator, select **Add Administrator**. Enter the name of the administrator that you want to add in the **Active Directory Name** field. Make sure to also provide the associated user account domain name. For example, **Domain** \ **UserName**.
3. Select the account you want to add and select **Add**. The new account should now appear in the list of server administrators.

Remove an administrator using the Management Console

1. Open the Microsoft Application Virtualization (App-V) Management Console and select **Administrators** in the navigation pane. The navigation pane displays a list of AD users and groups that currently have administrative access to the Microsoft Application Virtualization (App-V) server.
2. Right-click the account to be removed from the list of administrators and select **Remove**.

Related topics

- Operations for App-V

How to Register and Unregister a Publishing Server by Using the Management Console

5/31/2019 • 2 minutes to read • Edit Online

Applies to

- Windows 10, version 1607

You can register and unregister publishing servers that will synchronize with the App-V management server. You can also see the last attempt that the publishing server made to synchronize the information with the management server.

Use the following procedure to register or unregister a publishing server.

To register a publishing server using the Management Console

1. Connect to the Management Console and select **Servers**. For more information about how to connect to the Management Console, see How to Connect to the Management Console.
2. A list of publishing servers that already synchronize with the management server is displayed. Click Register New Server to register a new server.
3. Type a computer name of a domain joined computer on the **Server Name** line, to specify a name for the server. You should also include a domain name, for example, **MyDomain\TestServer**. Click **Check**.
4. Select the computer and click **Add** to add the computer to the list of servers. The new server will be displayed in the list.

To unregister a publishing server using the Management Console

1. Connect to the Management Console and select **Servers**. For more information about how to connect to the Management Console, see How to Connect to the Management Console.
2. A list of publishing servers that synchronize with the management server is displayed.
3. To unregister the server, right-click the computer name and select the computer name and select **unregister server**.

For App-V issues, use the App-V TechNet Forum.

Related topics

Operations for App-V

How to create a custom configuration file by using the App-V Management Console

6/10/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

You can use a dynamic configuration to customize an App-V package for a specific user. However, you must first create the dynamic user configuration (.xml) file or the dynamic deployment configuration file before you can use the files. Creation of the file is an advanced manual operation. For general information about dynamic user configuration files, see About App-V dynamic configuration.

You can create a dynamic user configuration file with the App-V Management Console by following the steps in this article.

Create a dynamic user configuration file

1. Right-click the name of the package that you want to view and select **Edit active directory access** to view the configuration that is assigned to a given user group. Alternatively, select the package, and click **Edit**.
2. Using the list of **AD Entities with Access**, select the AD group that you want to customize. Select **Custom** from the drop-down list. A link named **Edit** will appear.
3. Select **Edit**. The Dynamic User Configuration assigned to the AD Group will appear.
4. Select **Advanced**, and then select **Export Configuration**. Enter a file name and select **Save**. Now you can edit the file to configure a package for a user.

> **NOTE**
>
> If you want to export a configuration while running on Windows Server, make sure to disable the IE Enhanced Security Configuration setting. If this setting is enalbed and set to block downloads, you won't be able to download anything from the App-V Server.

Related topics

- Operations for App-V

How to Transfer Access and Configurations to Another Version of a Package by Using the Management Console

5/31/2019 • 2 minutes to read • Edit Online

Applies to

- Windows 10, version 1607

Use the following procedure to transfer the access and default package configurations to another version of a package by using the management console.

To transfer access and configurations to another version of a package

1. To view the package that you want to configure, open the App-V Management Console. Select the package to which you will transfer the new configuration, right-click the package and select **transfer default configuration from** or **transfer access and configurations from**, depending on the configuration that you want to transfer.

2. To transfer the configuration, in the **Select Previous Version** dialog box, select the package that contains the settings that you want to transfer, and then click **OK**.

   If you select **transfer default configuration from**, then only the underlying dynamic deployment configuration will be transferred.

   If you select **transfer access and configurations from**, then all access permissions, as well as the configuration settings, will be copied.

For App-V issues, use the App-V TechNet Forum.

Related topics

Operations for App-V

How to customize virtual applications extensions for a specific AD group by using the Management Console

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Use the following procedure to customize the virtual application extensions for an Active Directory (AD) group.

Customize virtual applications extensions for an AD group

1. To view the package that you want to configure, open the App-V Management Console. To view the configuration assigned to a given user group, select the package, then right-click the package name and select **Edit active directory access**. Alternatively, select the package and select **EDIT** in the **AD ACCESS** pane.

2. To customize an AD group, you can find the group from the list of **AD Entities with Access**. Then, using the drop-down box in the **Assigned Configuration** pane, select **Custom**, and then select **EDIT**.

3. To disable all extensions for a given application, clear **ENABLE**.

   To add a new shortcut for the selected application, right-click the application in the **SHORTCUTS** pane, and select **Add new shortcut**. To remove a shortcut, right-click the application in the **SHORTCUTS** pane and select **Remove Shortcut**. To edit an existing shortcut, right-click the application and select **Edit Shortcut**.

4. To view any other application extensions, select **Advanced**, and select **Export Configuration**. Enter a filename and select **Save**. You can view all application extensions that are associated with the package using the configuration file.

5. To edit additional application extensions, modify the configuration file and select **Import and Overwrite this Configuration**. Select the modified file and select **Open**. In the dialog, select **Overwrite** to complete the process.

Related topics

- Operations for App-V

How to View and Configure Applications and Default Virtual Application Extensions by Using the Management Console

5/31/2019 • 2 minutes to read • Edit Online

Applies to

- Windows 10, version 1607

Use the following procedure to view and configure default package extensions.

To view and configure default virtual application extensions

1. To view the package that you want to configure, open the App-V Management Console. Select the package that you want to configure, right-click the package name and select **edit default configuration**.
2. To view the applications contained in the specified package, in the **Default Configuration** pane, click **Applications**. To view the shortcuts for that package, click **Shortcuts**. To view the file type associations for that package, click **File Types**.
3. To enable the application extensions, select **ENABLE**.

   To enable shortcuts, select **ENABLE SHORTCUTS**. To add a new shortcut for the selected application, right-click the application in the **SHORTCUTS** pane and select **Add new shortcut**. To remove a shortcut, right-click the application in the **SHORTCUTS** pane and select **Remove Shortcut**. To edit an existing shortcut, right-click the application and select **Edit Shortcut**.
4. To view any other application extensions, click **Advanced** and click **Export Configuration**. Type in a filename and click **Save**. You can view all application extensions associated with the package using the configuration file.
5. To edit other application extensions, modify the configuration file and click **Import and Overwrite this Configuration**. Select the modified file and click **Open**. In the dialog box, click **Overwrite** to complete the process.

For App-V issues, use the App-V TechNet Forum.

Related topics

Operations for App-V

Managing Connection Groups

6/6/2019 • 2 minutes to read • Edit Online

Applies to

- Windows 10, version 1607

Connection groups enable the applications within a package to interact with each other in the virtual environment, while remaining isolated from the rest of the system. By using connection groups, administrators can manage packages independently and can avoid having to add the same application multiple times to a client computer.

Note In some previous versions of App-V, connection groups were referred to as Dynamic Suite Composition.

In this section:

About the Connection Group Virtual Environment	Describes the connection group virtual environment.
About the Connection Group File	Describes the connection group file.
How to Create a Connection Group	Explains how to create a new connection group.
How to Create a Connection Group with User-Published and Globally Published Packages	Explains how to create a new connection group that contains a mix of packages that are published to the user and published globally.
How to Delete a Connection Group	Explains how to delete a connection group.
How to Publish a Connection Group	Explains how to publish a connection group.
How to Make a Connection Group Ignore the Package Version	Explains how to configure a connection group to accept any version of a package, which simplifies package upgrades and reduces the number of connection groups you need to create.
How to Allow Only Administrators to Enable Connection Groups	Explains how to configure the App-V client so that only administrators (not end users) can enable or disable connection groups.

For App-V issues, use the App-V TechNet Forum.

Other resources for App-V connection groups

- Operations for App-V

About the connection group virtual environment

6/6/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

How package priority is determined

The virtual environment and its current state are associated with the connection group, not with the individual packages. If you remove an App-V package from the connection group, the state that existed as part of the connection group will not migrate with the package.

If the same package is a part of two different connection groups, you have to indicate which connection group App-V should use. For example, you might have two packages in a connection group that each define the same registry DWORD value.

The connection group that is used is based on the order in which a package appears inside the **AppConnectionGroup** XML document:

- The first package has the highest precedence.
- The second package has the second highest precedence.

Consider the following example section:

```
<appv:Packages>
  <appv:Package
    PackageId="A8731008-4523-4713-83A4-CD1363907160"
    VersionId="E889951B-7F30-418B-A69C-B37283BC0DB9"
  />
  <appv:Package
    PackageId="1DC709C8-309F-4AB4-BD47-F75926D04276"
    VersionId="01F1943B-C778-40AD-BFAD-AC34A695DF3C"
  />
  <appv:Package
    PackageId="04220DCA-EE77-42BE-A9F5-96FD8E8593F2"
    VersionId="E15EFFE9-043D-4C01-BC52-AD2BD1E8BAFA"
  />
</appv:Packages>
```

Assume that same DWORD value ABC (HKEY_LOCAL_MACHINE\software\contoso\finapp\region) is defined in the first and third package.

For this example, the DWORD value definition would be the following:

- Package 1 (A8731008-4523-4713-83A4-CD1363907160): HKEY_LOCAL_MACHINE\software\contoso\finapp\region=5
- Package 3 (04220DCA-EE77-42BE-A9F5-96FD8E8593F2): HKEY_LOCAL_MACHINE\software\contoso\finapp\region=10

Since Package 1 appears first, the AppConnectionGroup's virtual environment will have the single DWORD value of 5 (HKEY_LOCAL_MACHINE\software\contoso\finapp\region=5). This means that the virtual applications in Package 1, Package 2, and Package 3 will all see the value 5 when they query for HKEY_LOCAL_MACHINE\software\contoso\finapp\region.

Other virtual environment resources are resolved in a similar way, but usually collisions occur in the registry.

Merging identical package paths into one virtual directory in connection groups

If two or more packages in a connection group contain identical directory paths, the paths are merged into a single virtual directory inside the connection group's virtual environment. Merging these paths allows an application in one package to access files that are in a different package.

When you remove a package from a connection group, the removed package's applications can no longer access files from packages in the connection group it was removed from.

App-V looks up a file's name in the connection group in the order App-V packages are listed in the connection group manifest file.

The following example shows the order and relationship of a file name lookup in a connection group for **Package A** and **Package B**.

PACKAGE A	PACKAGE B
C:\Windows\System32	C:\Windows\System32
C:\AppTest	C:\AppTest

When a virtualized application tries to find a specific file, App-V will first for a matching file path in Package A. If it doesn't find a matching path in Package A, it will then search Package B using the following mapping rules:

- If a file named **test.txt** exists in the same virtual folder hierarchy in both application packages, App-V will use the first matching file.
- If a file named **bar.txt** exists in the virtual folder hierarchy of one application package, but not in the other, App-V will use the first matching file.

Related topics

- Managing Connection Groups

About the connection group file

6/10/2019 • 5 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Connection group file overview

What is a connection group?

A connection group is an App-V feature that can group packages together to create a virtual environment where applications within that package group can interact with each other.

For example, let's say you want to use plug-ins with Microsoft Office. You can create one package that contains the plug-ins and another package that contains Office, and then add both packages to the same connection group to enable Office to use those plug-ins.

How a connection group file works

When you apply an App-V connection group file, all packages specified in the file will be combined at runtime into a single virtual environment. Use the Microsoft Application Virtualization (App-V) connection group file to configure existing App-V connection groups.

An example file path for a package file would be %APPDATA%\Microsoft\AppV\Client\Catalog\PackageGroups{6CCC7575-162E-4152-9407-ED411DA138F4}{4D1E16E1-8EF8-41ED-92D5-8910A8527F96}.

Structure of the connection group XML file

This section will tell you more about the components of the connection group XML file.

Parameters that define the connection group

The following table describes the parameters in the XML file that define the connection group itself, not the packages.

FIELD	DESCRIPTION
Schema name	Name of the schema. If you want to use the "optional packages" and "use any version" features described in this table, you must specify the following schema in the XML file: `xmlns="https://schemas.microsoft.com/appv/2014/virtualapplicationconnect`
AppConnectionGroupId	Unique GUID identifier for this connection group. The connection group state is associated with this identifier. Specify this identifier only when you create the connection group. You can create a new GUID by entering **[Guid]::NewGuid()**.
VersionId	Version GUID identifier for this version of the connection group. When you update a connection group (for example, by adding or updating a new package), you must update the version GUID to reflect the new version.
DisplayName	Display name of the connection group.
Priority	Optional priority field for the connection group. A value of **0** indicates the highest priority. If a priority is required but has not been configured, the package will fail because it can't determine the correct connection group to use.

Parameters that define the packages in the connection group

In the **<Packages>** section of the connection group XML file, you list the member packages in the connection group by specifying each package's unique package identifier and version identifier, as described in the following table. The first package in the list has the highest precedence.

FIELD	DESCRIPTION
PackageId	Unique GUID identifier for this package. This GUID doesn't change when newer versions of the package are published.
VersionId	Unique GUID identifier for the version of the package. If you specify "*" for the package version, the GUID of the latest available package version is dynamically inserted.
IsOptional	Parameter that enables you to make a package optional within the connection group. Valid entries are: - "**true**"—package is optional in the connection group - "**false**"—package is required in the connection group

App-V example connection group XML file

The following example connection group XML file shows examples of the fields listed in the previous tables.

```
<?xml version="1.0" encoding="UTF-16"?>
<appv:AppConnectionGroup
xmlns="http://schemas.microsoft.com/appv/2014/virtualapplicationconnectiongroup"
xmlns:appv="http://schemas.microsoft.com/appv/2014/virtualapplicationconnectiongroup"
  AppConnectionGroupId="61BE9B14-D2B4-41CE-A6E3-A1B658DE7000"
  VersionId="E6B6AA57-F2A7-49C9-ADF8-F2B5B3C8A42F"
  Priority="0"
  DisplayName="Sample Connection Group">
  <appv:Packages>
    <appv:Package
      PackageId="1DC709C8-309F-4AB4-BD47-F75926D04276"
      VersionId="*"
      IsOptional="true"
    />
    <appv:Package
      PackageId="04220DCA-EE77-42BE-A9F5-96FD8E8593F2"
      VersionId="E15EFFE9-043D-4C01-BC52-AD2BD1E8BAFA"
      IsOptional="false"
    />
  </appv:Packages>
```

Configuring the priority of packages in a connection group

Package precedence is configured using the package list order. The first package in the document has the highest precedence. Subsequent packages in the list have descending priority.

Package precedence is the resolution for otherwise inevitable resource collisions during virtual environment initialization. For example, if two packages that are opening in the same virtual environment define the same registry DWORD value, the package with the highest precedence determines the value that is set.

You can use the connection group file to configure each connection group by using the following methods:

- Specify runtime priorities for connection groups. To edit priority by using the App-V Management Console, select the connection group and then select **Edit**.

  > **NOTE**
  > A package only requires priority if it's associated with more than one connection group.

- Specify package precedence within the connection group.

The priority field is required when a running virtual application initiates from a native application request, such as Microsoft Windows Explorer. The App-V client uses the priority to determine which connection group virtual environment the application should run in. This situation occurs if a virtual application is part of multiple connection groups.

If a virtual application is opened using another virtual application, the client will use the original virtual application's virtual environment. The priority field is not used in this case.

The following is an example of priority configuration:

The virtual application Microsoft Outlook is running in virtual environment **XYZ**. When you open an attached Microsoft Word

document, a virtualized version Microsoft Word opens in the virtual environment **XYZ**, regardless of the virtualized Microsoft Word's associated connection groups or runtime priorities.

Supported virtual application connection configurations

App-V supports the following application connection configurations.

- **An .exe file and plug-in (.dll)**. For example, you might want to distribute Microsoft Office to all users, but only distribute a Microsoft Excel plug-in to a small subset of those users.

  Enable the connection group for the appropriate users. Update each package individually as required.

- **An .exe file and a middleware application**. This is for cases where you have an application that requires a middleware application, or several applications that all depend on the same middleware runtime version.

  All computers that require one or more of the applications receive the connection groups with the application and middleware application runtime. You can optionally combine multiple middleware applications into a single connection group.

EXAMPLE	EXAMPLE DESCRIPTION
Virtual application connection group for the financial division	- Middleware application 1 - Middleware application 2 - Middleware application 3 - Middleware application runtime
Virtual application connection group for HR division	- Middleware application 5 - Middleware application 6 - Middleware application runtime

- **An. exe file and an .exe file**. This is for cases where you have an application that relies on another application, but you want to keep the packages separate for operational efficiencies, licensing restrictions, or rollout timelines.

  For example, if you are deploying Microsoft Lync 2010, you can use three packages:

  - Microsoft Office 2010
  - Microsoft Communicator 2007
  - Microsoft Lync 2010

  You can manage the deployment with the following connection groups:

  - Microsoft Office 2010 and Microsoft Communicator 2007
  - Microsoft Office 2010 and Microsoft Lync 2010

  After deployment, you can either create a single new Microsoft Office 2010 + Microsoft Lync 2010 package or keep and maintain them as separate packages and deploy them with a connection group.

Related topics

- Managing connection groups

How to create a connection group

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Use these steps to create a connection group by using the App-V Management Console. To use Windows PowerShell to create connection groups, see How to manage connection groups on a stand-alone computer by using Windows PowerShell.

When you place packages in a connection group, their package root paths merge. If you remove packages, only the remaining packages maintain the merged root.

Create a connection group

1. In the App-V Management Console, select **CONNECTION GROUPS** to display the Connection Groups library.
2. Select **ADD CONNECTION GROUP** to create a new connection group.
3. In the **New Connection Group** pane, enter a description for the group.
4. Select **EDIT** in the **CONNECTED PACKAGES** pane to add a new application to the connection group.
5. In the **PACKAGES Entire Library** pane, select the application to be added, then select the arrow to add the application.

   To remove an application, select the application to be removed in the **PACKAGES IN** pane and select the arrow.

   To reprioritize the applications in your connection group, use the arrows in the **PACKAGES IN** pane.

   > **IMPORTANT**
   > By default, the Active Directory Domain Services access configurations that are associated with a specific application are not added to the connection group. To transfer the Active Directory access configuration, select **ADD PACKAGE ACCESS TO GROUP ACCESS**, which is located in the **PACKAGES IN** pane.

6. After adding all the applications and configuring Active Directory access, select **Apply**.

Related topics

- Operations for App-V
- Managing connection groups

How to create a connection croup with user-published and globally published packages

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

You can create user-entitled connection groups that contain both user-published and globally published packages, using either of the following methods:

- How to use Windows PowerShell cmdlets to create user-entitled connection groups
- How to use the App-V Server to create user-entitled connection groups

Unsupported scenarios and potential issues

Here are some important things to know before you get started:

- If you add user-published packages in globally entitled connection groups, the connection group will fail.
- Track the connection groups where you've used a non-optional package before removing it with the **Unpublish-AppvClientPackage <package> -global** cmdlet.

  In situations where you have a globally published package that's listed as non-optional in a user-published connection group that also appears in other packages, running **Unpublish-AppvClientPackage <package> -global** cmdlet can unpublish the package from every connection group containing that package. Tracking connection groups can help you avoid unintentionally unpublishing non-optional packages.

How to use Windows PowerShell cmdlets to create user-entitled connection groups

1. Add and publish packages by using the following commands:

```
Add-AppvClientPackage <Package1_AppV_file_Path>
Add-AppvClientPackage <Package2_AppV_file_Path>
Publish-AppvClientPackage -PackageId <Package1_ID> -VersionId <Package1_Version_ID> -Global
Publish-AppvClientPackage -PackageId <Package2_ID> -VersionId <Package2_Version_ID>
```

2. Create the connection group XML file. For more information, see About the connection group file.
3. Add and publish the connection group by using the following commands:

```
Add-AppvClientConnectionGroup <Connection_Group_XML_file_Path>
Enable-AppvClientConnectionGroup -GroupId <CG_Group_ID> -VersionId <CG_Version_ID>
```

How to use the App-V Server to create user-entitled connection groups

1. Open the App-V Management Console.

2. Follow the instructions in How to publish a package by using the Management Console to publish packages globally and to the user.
3. Follow the instructions in How to create a connection group to create the connection group and add the user-published and globally published packages.

Related topics

- Managing Connection Groups

How to delete a connection group

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Use the following procedure to delete an existing App-V connection group.

Delete a connection group

1. Open the App-V Management Console and select **CONNECTION GROUPS**.
2. Right-click the connection group to be removed and select **delete**.

Related topics

- Operations for App-V
- Managing connection groups

How to Publish a Connection Group

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

After you create a connection group, you must publish it to computers that run the App-V client.

Publish a connection group

1. Open the App-V Management Console and select **CONNECTION GROUPS**.
2. Right-click the connection group to be published, and select **publish**.

Related topics

- Operations for App-V
- Managing connection groups

How to make a connection group ignore the package version

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

You can use Application Virtualization (App-V) to configure a connection group to use any version of a package, simplifying package upgrades and reducing the number of connection groups you need to create.

You can also configure a connection group to accept any version of a package, so that you can upgrade the package without having to disable the connection group.

- If the connection group has access to multiple versions of a package, App-V will use the latest version.
- If the connection group contains an optional package with an incorrect version, App-V ignores the package and won't block the connection group's virtual environment from being created.
- If the connection group contains a non-optional package that has an incorrect version, App-V won't be able to create the connection group's virtual environment.

Make a connection group ignore the package version with the App-V Server Management Console

1. In the Management Console, select **Connection Groups**.
2. Select the correct connection group from the Connection Groups library.
3. Select **Edit** in the Connected Packages pane.
4. Select the **Use Any Version** check box next to the package name, then select **Apply**.

For more about adding or upgrading packages, see How to add or upgrade packages by using the Management Console.

Make a connection group ignore the package version from the App-V client on a stand-alone computer

1. Create the connection group XML document.
2. Set the **Package** tag attribute **VersionID** to an asterisk (*****) to upgrade the package.
3. Enter the following cmdlet (including the path to the connection group XML document) to add the connection group:

   ```
   Add-AppvClientConnectionGroup
   ```

   For more information about how to use the **Add-AppvClientConnectionGroup** cmdlet, see **Add-AppvClientConnectionGroup**.
4. When you upgrade a package, use the following cmdlets to remove the old package, add the upgraded package, and publish the upgraded package:

- **Remove-AppvClientPackage**
- **Add-AppvClientPackage**
- **Publish-AppvClientPackage**

For more information, see How to manage App-V packages running on a stand-alone computer by using Windows PowerShell.

Related topics

- Managing connection groups

How to allow only administrators to enable connection groups

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

You can configure the App-V client so that only administrators, not users, can enable or disable connection groups. In earlier versions of App-V, there was no way to restrict access to disabling connection groups to users.

> **NOTE**
>
> This feature is supported starting in App-V 5.0 SP3.

Use one of the following methods to allow only administrators to enable or disable connection groups.

METHOD	STEPS
Group Policy setting	Enable the "Require publish as administrator" Group Policy setting, which is located in the following Group Policy Object node: **Computer Configuration** > **Administrative Templates** > **System** > **App-V** > **Publishing**
Windows PowerShell cmdlet	Run the **Set-AppvClientConfiguration** cmdlet with the -*RequirePublishAsAdmin* parameter. Parameter values: - **0** – False - **1** – True Example: `Set-AppvClientConfiguration -RequirePublishAsAdmin 1`

Related topics

- Managing Connection Groups

Deploying App-V packages by using electronic software distribution (ESD)

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

You can deploy App-V packages using an electronic software distribution (ESD) solution. For information about planning to deploy App-V packages with an ESD, see Planning to deploy App-V with an electronic software distribution system.

To learn how to deploy App-V packages with Microsoft System Center 2012 Configuration Manager, see Introduction to application management in Configuration Manager

How to deploy virtualized packages using an ESD

To learn more about how to deploy virtualized packages using an ESD, see How to deploy App-V packages using electronic software distribution.

How to enable only administrators to publish packages by using an ESD

To learn how to configure the App-V client to enable only administrators to publish and unpublish packages when you're using an ESD, see How to enable only administrators to publish packages by using an ESD.

Related topics

- App-V and Citrix integration
- Operations for App-V

How to deploy App-V packages using electronic software distribution

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

You can use an electronic software distribution (ESD) system to deploy App-V virtual applications to App-V clients.

For component requirements and options for using an ESD to deploy App-V packages, see Planning to deploy App-V with an electronic software distribution system.

Use one of the following methods to publish packages to App-V client computers with an ESD:

- Use the functionality in a third-party ESD.
- Install the application on the target client computer with the associated Windows Installer (.msi) file that's created when you initially sequence the application. The .msi file contains the associated App-V package file information used to configure a package and copies the required package files to the client.
- Use Windows PowerShell cmdlets to deploy virtualized applications. For more information about using Windows PowerShell and App-V, see Administering App-V by using Windows PowerShell.

METHOD	DESCRIPTION
Functionality provided by a third-party ESD	Use the functionality in a third-party ESD.
Stand-alone Windows Installer	Install the application on the target client computer by using the associated Windows Installer (.msi) file that is created when you initially sequence an application. The Windows Installer file contains the associated App-V package file information used to configure a package and copies the required package files to the client.
Windows PowerShell	Use Windows PowerShell cmdlets to deploy virtualized applications. For more information about using Windows PowerShell and App-V, see Administering App-V by using Windows PowerShell.

Deploy App-V packages with an ESD

1. Install the App-V Sequencer on a computer in your environment. For more information about installing the sequencer, see How to install the Sequencer.
2. Use the App-V Sequencer to create a virtual application. To learn more about creating virtual applications, see Creating and managing App-V virtualized applications.
3. After you create the virtual application, deploy the package by using your ESD solution.

Related topics

- Operations for App-V

How to enable only administrators to publish packages by using an ESD

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Starting in App-V 5.0 SP3, you can configure the App-V client so that only administrators (not end users) can publish or unpublish packages. In earlier versions of App-V, you could not prevent end users from performing these tasks.

Here's how to enable only administrators to publish or unpublish packages:

1. Navigate to the following Group Policy Object node:

   Computer Configuration > **Administrative Templates** > **System** > **App-V** > **Publishing**.

2. Enable the **Require publish as administrator** Group Policy setting.

   To instead use Windows PowerShell to set this item, see Understanding pending packages: UserPending and GlobalPending.

Using the App-V Client Management Console

6/6/2019 • 2 minutes to read • Edit Online

Applies to

- Windows 10, version 1607

This topic provides information about using the Application Virtualization (App-V) client management console to manage packages on the computer running the App-V client.

Obtain the client management console

The client management console is separate from the App-V client itself. You can download the client management console from the Microsoft Download Center.

> **NOTE**
> To perform all of the actions available using the client management console, you must have administrative access on the computer running the App-V client.

Options for managing the App-V client

The App-V client has associated settings that can be configured to determine how the client will run in your environment. You can manage these settings on the computer that runs the client, or you can use Windows PowerShell or Group Policy. For more information about configuring the client by using Windows PowerShell or Group Policy, see:

- Deploying the App-V Sequencer and Configuring the Client
- How to Modify Client Configuration by Using Windows PowerShell
- How to Configure the Client to Receive Package and Connection Groups Updates From the Publishing Server

The App-V client management console

You can obtain information about the App-V client or perform specific tasks by using the App-V client management console. Many of the tasks that you can perform in the client management console you can also perform by using Windows PowerShell. The associated Windows PowerShell cmdlets for each action are also displayed in the following table. For more information about how to use Windows PowerShell, see Administering App-V by Using Windows PowerShell.

The client management console contains the following described main tabs.

TAB	DESCRIPTION

TAB	DESCRIPTION
Overview	The **Overview** tab contains the following elements: • Update – Use the **Update** tile to refresh a virtualized application or to receive a new virtualized package. The **Last Refresh** displays the current version of the virtualized package. • Download all virtual applications – Use the **Download** tile to download all of the packages provisioned to the current user. (Associated Windows PowerShell cmdlet: **Mount-AppvClientPackage**) • Work Offline – Use this tile to disallow all automatic and manual virtual application updates. (Associated Windows PowerShell cmdlet: **Set-AppvPublishServer –UserRefreshEnabled –GlobalRefreshEnabled**)
Virtual Apps	The **VIRTUAL APPS** tab displays all of the packages that have been published to the user. You can also click a specific package and see all of the applications that are part of that package. This displays information about packages that are currently in use and how much of each package has been downloaded to the computer. You can also start and stop package downloads. Additionally, you can repair the user state. A repair will delete all user data that is associated with a package.
App Connection Groups	The **APP CONNECTION GROUPS** tab displays all of the connection groups that are available to the current user. Click a specific connection group to see all of the packages that are part of the selected group. This displays information about connection groups that are already in use and how much of the connection group contents have been downloaded to the computer. Additionally, you can start and stop connection group downloads. You can use this section to initiate a repair. A repair will remove all of the user state that is associated a connection group. (Associated Windows PowerShell cmdlets: Download - **Mount-AppvClientConnectionGroup**. Repair -**AppvClientConnectionGroup**.)

For App-V issues, use the App-V TechNet Forum.

Related topics

Operations for App-V

Automatically clean up unpublished packages on the App-V client

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1703

If you wanted to free up additional storage space in previous versions of App-V, you would have had to manually remove your unpublished packages from your client devices. Windows 10, version 1703 introduces the ability to use PowerShell or Group Policy settings to automatically clean up your unpublished packages after restarting your device.

Clean up with PowerShell cmdlets

You can enter PowerShell cmdlets to turn on the **AutoCleanupEnabled** setting, which will automatically clean up your unpublished App-V packages from your App-V client devices.

Turn on the AutoCleanupEnabled option

1. Open PowerShell as an admin and enter the following cmdlet to turn on the automatic package cleanup functionality:

   ```
   Set-AppvClientConfiguration -AutoCleanupEnabled 1
   ```

   After running the cmdlet, you should see the following info on the PowerShell screen:

NAME	VALUE	SETBYGROUPPOLICY
AutoCleanupEnabled	1	False

2. Run the following cmdlet to check if the configuration has the cleanup setting turned on.

   ```
   Get-AppvClientConfiguration
   ```

   If the **AutoCleanupEnabled** option shows a value of **1** in the configuration list, that means the setting is turned on.

Clean up with Group Policy settings

Using Group Policy, you can turn on the **Enable automatic cleanup of unused App-V packages** setting to automatically clean up your unpublished App-V packages from your App-V client devices.

Turn on the Enable automatic cleanup of unused App-V packages setting

1. Open your Group Policy editor and select the **Administrative Templates\System\App-V\PackageManagement\Enable automatic cleanup of unused App-V packages** setting.

2. Select **Enabled**, then select **OK**.

   After your Group Policy updates and you reset the client, the setting will clean up any unpublished App-V packages on the App-V client.

Related topics

- Download the Windows ADK
- Download the Microsoft Application Virtualization 5.0 Client UI Application
- Using the App-V Client Management Console

Migrating to App-V from previous versions

6/6/2019 • 4 minutes to read • Edit Online

Applies to

- Windows 10, version 1607

To migrate from App-V 4.x to App-V for Windows 10, you must upgrade to App-V 5.x first.

Improvements to the App-V Package Converter

You can now use the package converter to convert App-V 4.6 packages that contain scripts, and registry information and scripts from source .osd files are now included in package converter output.

You can also use the `-OSDsToIncludeInPackage` parameter with the `ConvertFrom-AppvLegacyPackage` cmdlet to specify which .osd files' information is converted and placed within the new package.

NEW IN APP-V FOR WINDOWS 10	PRIOR TO APP-V FOR WINDOWS 10
New .xml files are created corresponding to the .osd files associated with a package; these files include the following information: • environment variables • shortcuts • file type associations • registry information • scripts You can now choose to add information from a subset of the .osd files in the source directory to the package using the `-OSDsToIncludeInPackage` parameter.	Registry information and scripts included in .osd files associated with a package were not included in package converter output. The package converter would populate the new package with information from all of the .osd files in the source directory.

Example conversion statement

To understand the new process, review the following example `ConvertFrom-AppvLegacyPackage` package converter statement.

If the source directory (\\OldPkgStore\ContosoApp) includes the following:

- ContosoApp.sft
- ContosoApp.msi
- ContosoApp.sprj
- ContosoApp_manifest.xml
- X.osd
- Y.osd
- Z.osd

And you run this command:

```
ConvertFrom-AppvLegacyPackage –SourcePath \\OldPkgStore\ContosoApp\
-DestinationPath \\NewPkgStore\ContosoApp\
-OSDsToIncludeInPackage X.osd,Y.osd
```

The following is created in the destination directory (\\NewPkgStore\ContosoApp):

- ContosoApp.appv
- ContosoApp.msi
- ContosoApp_DeploymentConfig.xml
- ContosoApp_UserConfig.xml
- X_Config.xml
- Y_Config.xml
- Z_Config.xml

In the above example:

THESE SOURCE DIRECTORY FILES...	...ARE CONVERTED TO THESE DESTINATION DIRECTORY FILES...	...AND WILL CONTAIN THESE ITEMS	DESCRIPTION
• X.osd • Y.osd • Z.osd	• X_Config.xml • Y_Config.xml • Z_Config.xml	• Environment variables • Shortcuts • File type associations • Registry information • Scripts	Each .osd file is converted to a separate, corresponding .xml file that contains the items listed here in App-V deployment configuration format. These items can then be copied from these .xml files and placed in the deployment configuration or user configuration files as desired. In this example, there are three .xml files, corresponding with the three .osd files in the source directory. Each .xml file contains the environment variables, shortcuts, file type associations, registry information, and scripts in its corresponding .osd file.

THESE SOURCE DIRECTORY FILES...	...ARE CONVERTED TO THESE DESTINATION DIRECTORY FILES...	...AND WILL CONTAIN THESE ITEMS	DESCRIPTION
• X.osd • Y.osd	• ContosoApp.appv • ContosoApp_DeploymentConfig.xml • ContosoApp_UserConfig.xml	• Environment variables • Shortcuts • File type associations	The information from the .osd files specified in the `-OSDsToIncludeInPackage` parameter are converted and placed inside the package. The converter then populates the deployment configuration file and the user configuration file with the contents of the package, just as App-V Sequencer does when sequencing a new package. In this example, environment variables, shortcuts, and file type associations included in X.osd and Y.osd were converted and placed in the App-V package, and some of this information was also included in the deployment configuration and user configuration files. X.osd and Y.osd were used because they were included as arguments to the `-OSDsToIncludeInPackage` parameter. No information from Z.osd was included in the package, because it was not included as one of these arguments.

Converting packages created using a prior version of App-V

Use the package converter utility to upgrade virtual application packages created using versions of App-V prior to App-V 5.0. The package converter uses Windows PowerShell to convert packages and can help automate the process if you have many packages that require conversion. App-V packages created with App-V 5.x don't need to be converted.

Important After you convert an existing package you should test the package prior to deploying the package to ensure the conversion process was successful.

What to know before you convert existing packages

ISSUE	WORKAROUND
Virtual packages using DSC are not linked after conversion.	Link the packages using connection groups. See Managing Connection Groups.
Environment variable conflicts are detected during conversion.	Resolve any conflicts in the associated **.osd** file.
Hard-coded paths are detected during conversion.	Hard-coded paths are difficult to convert correctly. The package converter will detect and return packages with files that contain hard-coded paths. View the file with the hard-coded path, and determine whether the package requires the file. If so, it is recommended to re-sequence the package.

When converting a package check for failing files or shortcuts, locate the item in App-V 4.6 package. It could possibly be a hard-coded path. Convert the path.

Note It is recommended that you use the App-V sequencer for converting critical applications or applications that need to take advantage of features. See How to Sequence a New Application with App-V.

If a converted package does not open after you convert it, it is also recommended that you re-sequence the application using the App-V sequencer.

How to Convert a Package Created in a Previous Version of App-V

Migrating the App-V Server Full Infrastructure

There is no direct method to upgrade to a full App-V infrastructure. Use the information in the following section for information about upgrading the App-V server.

TASK	MORE INFORMATION
Review prerequisites.	App-V Server prerequisite software.
Enable the App-V client.	Enable the App-V desktop client.
Install App-V Server.	How to Deploy the App-V Server.
Migrate existing packages.	See Converting packages created using a prior version of App-V earlier in this topic.

For App-V issues, use the App-V TechNet Forum.

Other resources for performing App-V migration tasks

- Operations for App-V
- A simplified Microsoft App-V 5.1 Management Server upgrade procedure

How to convert a package created in a previous version of App-V

6/10/2019 • 3 minutes to read • Edit Online

Applies to: Windows 10, version 1607

You can use the package converter utility to upgrade virtual application packages created by previous versions of App-V. This section will tell you how to convert existing virtual application packages for upgrade.

> **NOTE**
> If you are running a computer with a 64-bit architecture, you must use the x86 version of Windows PowerShell.

The package converter can only directly convert packages created by an App-V sequencer version 4.5 or later. Packages created with an App-V version earlier than 4.5 must be upgraded to at least App-V 4.5 before conversion.

> **IMPORTANT**
> In order to keep your files secure, you must configure the package converter to always save the package ingredients file to a secure location and directory that can only be accessed by an administrator. When you deploy the package, you should either save the package to a secure location or make sure that no other users can sign in during the conversion process.

App-V 4.6 installation folder is redirected to virtual file system root

When you convert packages from App-V 4.6 to App-V for Windows 10, the App-V for Windows 10 package can access the hardcoded drive that you were required to use when you created 4.6 packages. The drive letter will be the drive you selected as the installation drive on the 4.6 sequencing machine. (The default drive is drive Q.)

The App-V package converter will save the App-V 4.6 installation root folder and short folder names in the FilesystemMetadata.xml file in the **Filesystem** element. When the App-V for Windows 10 client creates the virtual process, it will map requests from the App-V 4.6 installation root to the virtual file system root.

Getting started

1. Install the App-V Sequencer on a computer in your environment. For information about how to install the Sequencer, see How to install the Sequencer.

2. You can enter the following cmdlets to check or convert packages:

   - **Test-AppvLegacyPackage**—This cmdlet checks packages. It will return information about any failures with the package such as missing **.sft** files, an invalid source, **.osd** file errors, or invalid package version. This cmdlet will not parse the **.sft** file or do any in-depth validation. For information about options and basic functionality for this cmdlet, using Windows PowerShell, enter the following cmdlet:

     ```
     Test-AppvLegacyPackage -?
     ```

- **ConvertFrom-AppvLegacyPackage**—This cmdlet converts packages from legacy versions to updated versions. To convert an existing package, enter the following cmdlet:

  ```
  ConvertFrom-AppvLegacyPackage C:\contentStore C:\convertedPackages
  ```

  In this cmdlet, `C:\contentStore` represents the location of the existing package and `C:\convertedPackages` is the output directory to which the resulting App-V for Windows 10 virtual application package file will be saved. By default, if you do not specify a new name, the old package name will be used.

  Additionally, the package converter optimizes performance of packages in App-V for Windows 10 by setting the package to stream fault the App-V package. This is more performant than the primary feature block and fully downloading the package. The flag **DownloadFullPackageOnFirstLaunch** allows you to convert the package and set the package to be fully downloaded by default.

NOTE

```
>Before you specify the output directory, you must create the output directory.
```

Advanced Conversion Tips

- Piping—Windows PowerShell supports piping. Piping allows you to enter cmdlets like this example:

  ```
  dir C:\contentStore\myPackage | Test-AppvLegacyPackage
  ```

  In this example, the directory object that represents `myPackage` will be given as input to the **Test-AppvLegacyPackage** cmdlet and bound to the *-Source* parameter. Piping like this is especially useful when you want to batch commands together, such as in the following example cmdlet:

  ```
  dir .\ | Test-AppvLegacyPackage | ConvertFrom-AppvLegacyAppvPackage -Target .\ConvertedPackages
  ```

  This piped example command tests packages, then passes the objects on for conversion. You can also apply a filter on packages without errors or only specify a directory which contains an **.sprj** file or pipe them to another cmdlet that adds the filtered package to the server or publishes them to the App-V client.

- Batching—The Windows PowerShell command enables batching. More specifically, the cmdlets support taking a string[] object for the *-Source* parameter that represents a list of directory paths. This allows you to enter the following cmdlets together:

  ```
  $packages = dir C:\contentStore
  ConvertFrom-AppvLegacyAppvPackage-Source $packages -Target C:\ConvertedPackages
  ```

  Alternatively, you can use piping like this:

  ```
  dir C:\ContentStore | ConvertFrom-AppvLegacyAppvPackage -Target C:\ConvertedPackages
  ```

- Other functionality—Windows PowerShell has other built-in functionality for features such as aliases, lazy-binding, .NET Object, and many others. These features can help you create advanced scenarios for the Package Converter.

Related topics

- Operations for App-V

Maintaining App-V

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

After you have deployed App-V for Windows 10, you can use the following information to maintain the App-V infrastructure.

Moving the App-V server

The App-V server connects to the App-V database, which means you can install the management component and connect it to the App-V database on any computer on the network. For more information, see How to move the App-V server to another computer.

Determine if an App-V application is running virtualized

Independent software vendors (ISV) who want to determine if an application is running virtualized with App-V should open a named object called **AppVVirtual-<PID>** in the default namespace (PID stands for process ID). To find the process ID of the process you're currently using, enter the Windows API **GetCurrentProcessId()**.

For example, let's say the process ID is 4052. If you can successfully open a named Event object called **AppVVirtual-4052** with the **OpenEvent()** API in the default read access namespace, then the application is virtual. If the **OpenEvent()** call fails, the application isn't virtual.

Additionally, ISVs who want to explicitly virtualize or not virtualize calls on specific APIs with App-V 5.1 and later can use the **VirtualizeCurrentThread()** and **CurrentThreadIsVirtualized()** functions implemented in the AppEntSubsystems32.dll module to hint to a downstream component whether the call should be virtualized or not.

Other resources for maintaining App-V

- Operations for App-V

How to move the App-V server to another computer

5/31/2019 • 2 minutes to read • Edit Online

Applies to

- Windows Server 2016

Use the following information to create a new management server console in your environment.

To create a new management server console

Follow these steps to create a new management server console:

1. Install the management server on a computer in your environment. For more information about installing the management server see Deploying the App-V server.
2. After you have completed the installation, use the following link to connect it to the App-V database - How to install the Management Server on a Standalone Computer and Connect it to the Database.

For App-V issues, use the App-V TechNet Forum.

Related topics

Operations for App-V

Administering App-V by using Windows PowerShell

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Microsoft Application Virtualization (App-V) supports Windows PowerShell cmdlets that give administrators a quick and easy way to manage App-V. The following sections will tell you more about how to use Windows PowerShell with App-V.

How to administer App-V with Windows PowerShell

The following table lists articles that will tell you more about how to use PowerShell for App-V.

NAME	DESCRIPTION
How to load the Windows PowerShell cmdlets for App-V and get cmdlet help	Describes how to install the Windows PowerShell cmdlets and find cmdlet help and examples.
How to manage App-V packages running on a stand-alone computer by using Windows PowerShell	Describes how to manage the client package lifecycle on a stand-alone computer with Windows PowerShell.
How to manage connection groups on a stand-alone computer by using Windows PowerShell	Describes how to manage connection groups with Windows PowerShell.
How to modify client configuration by using Windows PowerShell	Describes how to modify the client with Windows PowerShell.
How to apply the user configuration file by using Windows PowerShell	Describes how to apply a user configuration file with Windows PowerShell.
How to apply the deployment configuration file by using Windows PowerShell	Describes how to apply a deployment configuration file with Windows PowerShell.
How to sequence a package by using Windows PowerShell	Describes how to create a new package with Windows PowerShell.
How to create a package accelerator by using Windows PowerShell	Describes how to create a package accelerator with Windows PowerShell. You can use package accelerators to automatically sequence large, complex applications.
How to enable reporting on the App-V client by using Windows PowerShell	Describes how to enable the computer running the App-V Client to send reporting information.
How to install the App-V databases and convert the associated security identifiers by using Windows PowerShell	Describes how to take an array of account names and to convert each of them to the corresponding SID in standard and hexadecimal formats.
How to configure the client to receive package and connection groups updates from the publishing server	Describes how to use Windows PowerShell to configure a client after you have deployed the App-V management and publishing servers and added the required packages and connection groups.

> **IMPORTANT**
> Make sure that any script you execute with your App-V packages matches the execution policy that you have configured for Windows PowerShell.

Windows PowerShell error handling

The following table describes Windows PowerShell error handling for App-V.

EVENT	ACTION
Using the **RollbackOnError** attribute with embedded scripts	When you use the **RollbackOnError** attribute with embedded scripts, the attribute is ignored for the following events: - Removing a package - Unpublishing a package - Terminating a virtual environment - Terminating a process
Package name contains **$**	If a package name contains the character $, you must use a single-quote ('). For example: `Add-AppvClientPackage 'Contoso$App.appv'`

Related topics

- Operations for App-V

How to load the Windows PowerShell cmdlets for App-V and get cmdlet help

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

Requirements for using Windows PowerShell cmdlets

This section will tell you what you'll need to use the PowerShell cmdlets.

How to let users access PowerShell cmdlets

You can grant your users access to PowerShell cmdlets through one of the following methods:

- While you're deploying and configuring the App-V server, specify an Active Directory group or individual user with permissions to manage the App-V environment. For more information, see How to deploy the App-V Server.
- After you've deployed the App-V server, you can use the App-V Management console to add an additional Active Directory group or user. For more information, see How to add or remove an administrator by using the Management console.

Elevated command prompt

You'll need an elevated command prompt to run the following cmdlets:

- **Add-AppvClientPackage**
- **Remove-AppvClientPackage**
- **Set-AppvClientConfiguration**
- **Add-AppvClientConnectionGroup**
- **Remove-AppvClientConnectionGroup**
- **Add-AppvPublishingServer**
- **Remove-AppvPublishingServer**
- **Send-AppvClientReport**
- **Set-AppvClientMode**
- **Set-AppvClientPackage**
- **Set-AppvPublishingServer**

Other cmdlets

The following cmdlets are ones that end-users can run unless you configure them to require an elevated command prompt.

- **Publish-AppvClientPackage**
- **Unpublish-AppvClientPackage**

To configure these cmdlets to require an elevated command prompt, use one of the following methods:

- Run the **Set-AppvClientConfiguration** cmdlet with the *-RequirePublishAsAdmin* parameter. For more information, see the following resources:
  - How to manage connection groups on a stand-alone computer by using Windows PowerShell
  - Understanding pending packages: UserPending and GlobalPending
- Enable the **Require publish as administrator** Group Policy setting for App-V Clients. For more information,

see How to publish a package by using the Management Console.

Loading the Windows PowerShell cmdlets

To load the Windows PowerShell cmdlet modules:

1. Open Windows PowerShell or Windows PowerShell Integrated Scripting Environment (ISE).
2. Enter one of the following cmdlets to load a list of usable cmdlets for the module you want:

APP-V COMPONENT	CMDLET TO ENTER
App-V Server	**Import-Module AppvServer**
App-V Sequencer	**Import-Module AppvSequencer**
App-V Client	**Import-Module AppvClient**

Getting help for the Windows PowerShell cmdlets

Starting in App-V 5.0 SP3, cmdlet help is available in two formats:

- As a downloadable module in PowerShell. To access the module, open Windows PowerShell or Windows PowerShell Integrated Scripting Environment (ISE) and enter one of the cmdlets from the following table.

APP-V COMPONENT	CMDLET TO ENTER
App-V Server	**Update-Help -Module AppvServer**
App-V Sequencer	**Update-Help -Module AppvSequencer**
App-V Client	**Update-Help -Module AppvClient**

- Online in the Microsoft Desktop Optimization Pack.

Displaying the help for a Windows PowerShell cmdlet

To display help for a specific Windows PowerShell cmdlet:

1. Open Windows PowerShell or Windows PowerShell Integrated Scripting Environment (ISE).
2. Enter **Get-Help** followed by the cmdlet you need help with. For example:

```
Get-Help Publish-AppvClientPackage
```

How to manage App-V packages running on a stand-alone computer by using Windows PowerShell

6/10/2019 • 3 minutes to read • Edit Online

Applies to: Windows 10, version 1607

The following sections explain how to perform various management tasks on a stand-alone client computer with Windows PowerShell cmdlets.

Return a list of packages

Enter the **Get-AppvClientPackage** cmdlet to return a list of packages entitled to a specific user. Its parameters are *-Name*, *-Version*, *-PackageID*, and *-VersionID*.

For example:

```
Get-AppvClientPackage –Name "ContosoApplication" -Version 2
```

Add a package

Use the **Add-AppvClientPackage** cmdlet to add a package to a computer.

> **IMPORTANT**
> This example only adds a package. It does not publish the package to the user or the computer.

For example:

```
$Contoso = Add-AppvClientPackage \\\\path\\to\\appv\\package.appv
```

Publish a package

Use the **Publish-AppvClientPackage** cmdlet to publish a package that has been added to either a specific user or globally to any user on the computer.

Enter the cmdlet with the application name to publish it to the user.

```
Publish-AppvClientPackage "ContosoApplication"
```

To publish the application globally, just add the *-Global* parameter.

```
Publish-AppvClientPackage "ContosoApplication" -Global
```

Publish a package to a specific user

> **NOTE**
> You must use App-V 5.0 SP2 Hotfix Package 5 or later to use this parameter.

An administrator can publish a package to a specific user by specifying the optional *–UserSID* parameter with the **Publish-AppvClientPackage** cmdlet, where *-UserSID* represents the end user's security identifier (SID).

To use this parameter:

- You can run this cmdlet from the user or administrator session.
- You must be logged in with administrative credentials to use the parameter.
- The end user must be signed in.
- You must provide the end user's security identifier (SID).

For example:

```
Publish-AppvClientPackage "ContosoApplication" -UserSID S-1-2-34-56789012-3456789012-345678901-2345
```

Add and publish a package

Use the **Add-AppvClientPackage** cmdlet to add a package to a computer and publish it to the user.

For example:

```
Add-AppvClientPackage <path to App-V package> | Publish-AppvClientPackage
```

Unpublish an existing package

Use the **Unpublish-AppvClientPackage** cmdlet to unpublish a package which has been entitled to a user but not remove the package from the computer.

For example:

```
Unpublish-AppvClientPackage "ContosoApplication"
```

Unpublish a package for a specific user

> **NOTE**
> You must use App-V 5.0 SP2 Hotfix Package 5 or later to use this parameter.

An administrator can unpublish a package for a specific user by using the optional *-UserSID* parameter with the **Unpublish-AppvClientPackage** cmdlet, where *-UserSID* represents the end user's security identifier (SID).

To use this parameter:

- You can run this cmdlet from the user or administrator session.
- You must sign in with administrative credentials to use the parameter.
- The end user must be signed in.
- You must provide the end user's security identifier (SID).

For example:

```
Unpublish-AppvClientPackage "ContosoApplication" -UserSID S-1-2-34-56789012-3456789012-345678901-2345
```

Remove an existing package

Use the **Remove-AppvClientPackage** cmdlet to remove a package from the computer.

For example:

```
Remove-AppvClientPackage "ContosoApplication"
```

> **NOTE**
> App-V cmdlets have been assigned to variables for the previous examples for clarity only; assignment is not a requirement. Most cmdlets can be combined as displayed in Add and publish a package. For a detailed tutorial, see App-V 5.0 Client PowerShell Deep Dive.

Enable only administrators to publish or unpublish packages

Starting in App-V 5.0 SP3, you can use the **Set-AppvClientConfiguration** cmdlet and *-RequirePublishAsAdmin* parameter to enable only administrators (not end users) to publish or unpublish packages.

You can set the *-RequirePublishAsAdmin* parameter to the following values:

- 0: False
- 1: True

For example:

```
Set-AppvClientConfiguration –RequirePublishAsAdmin1
```

To use the App-V Management console to set this configuration, see How to publish a package by using the Management Console.

About pending packages: UserPending and GlobalPending

Starting in App-V 5.0 SP2, if you run a Windows PowerShell cmdlet that affects a package currently in use, the task you're trying to perform is placed in a pending state. For example, if you try to publish a package when an application in that package is being used, and then run **Get-AppvClientPackage**, the pending status appears in the cmdlet output as follows:

CMDLET OUTPUT ITEM	DESCRIPTION
UserPending	Indicates whether the listed package has a pending task that is being applied to the user: - True - False

CMDLET OUTPUT ITEM	DESCRIPTION
GlobalPending	Indicates whether the listed package has a pending task that is being applied globally to the computer: - True - False

The pending task will run later, according to the following rules:

TASK TYPE	APPLICABLE RULE
User-based (for example, publishing a package to a user)	The pending task will be performed after the user logs off and then logs back on.
Globally based (for example, enabling a connection group globally)	The pending task will be performed when the computer is shut down and then restarted.

For more information about pending tasks, see Upgrading an in-use App-V package.

Related topics

- Operations for App-V
- Administering App-V by using Windows PowerShell

How to Manage Connection Groups on a Stand-alone Computer by Using Windows PowerShell

5/31/2019 • 2 minutes to read • Edit Online

Applies to

- Windows 10, version 1607

An App-V connection group allows you to run all the virtual applications as a defined set of packages in a single virtual environment. For example, you can virtualize an application and its plug-ins by using separate packages, but run them together in a single connection group.

A connection group XML file defines the connection group for the App-V client. For information about the connection group XML file and how to configure it, see About the Connection Group File.

This topic explains the following procedures:

- To add and publish the App-V packages in the connection group
- To add and enable the connection group on the App-V client
- To enable or disable a connection group for a specific user
- To allow only administrators to enable connection groups

To add and publish the App-V packages in the connection group

1. To add and publish the App-V packages to the computer running the App-V client, type the following command:

   Add-AppvClientPackage –path c:\tmpstore\quartfin.appv | Publish-AppvClientPackage

2. Repeat **step 1** of this procedure for each package in the connection group.

To add and enable the connection group on the App-V client

1. Add the connection group by typing the following command:

   Add-AppvClientConnectionGroup –path c:\tmpstore\financ.xml

2. Enable the connection group by typing the following command:

   Enable-AppvClientConnectionGroup –name "Financial Applications"

   When any virtual applications that are in the member packages are run on the target computer, they will run inside the connection group's virtual environment and will be available to all the virtual applications in the other packages in the connection group.

To enable or disable a connection group for a specific user

1. Review the parameter description and requirements:

   - The parameter enables an administrator to enable or disable a connection group for a specific user.
   - You must use App-V 5.0 SP2 Hotfix Package 5 or later to use this parameter.

- You can run this cmdlet from the user or administrator session.
- You must be logged in with administrative credentials to use the parameter.
- The end user must be logged in.
- You must provide the end user's security identifier (SID).

2. Use the following cmdlets, and add the optional **–UserSID** parameter, where **-UserSID** represents the end user's security identifier (SID):

CMDLET	EXAMPLES
Enable-AppVClientConnectionGroup	Enable-AppVClientConnectionGroup "ConnectionGroupA" -UserSID S-1-2-34-56789012-3456789012-345678901-2345
Disable-AppVClientConnectionGroup	Disable-AppVClientConnectionGroup "ConnectionGroupA" -UserSID S-1-2-34-56789012-3456789012-345678901-2345

To allow only administrators to enable connection groups

1. Review the description and requirement for using this cmdlet:
   - Use this cmdlet and parameter to configure the App-V client to allow only administrators (not end users) to enable or disable connection groups.
   - You must be using at least App-V 5.0 SP3 to use this cmdlet.
2. Run the following cmdlet and parameter:

CMDLET	PARAMETER AND VALUES	EXAMPLE
Set-AppvClientConfiguration	-RequirePublishAsAdmin • 0 - False • 1 - True	Set-AppvClientConfiguration -RequirePublishAsAdmin 1

For App-V issues, use the App-V TechNet Forum.

Related topics

Operations for App-V

Administering App-V by Using Windows PowerShell

How to Modify Client Configuration by Using Windows PowerShell

5/31/2019 • 2 minutes to read • Edit Online

Applies to

- Windows 10, version 1607

Use the following procedure to configure the App-V client configuration.

1. To configure the client settings using Windows PowerShell, use the **Set-AppVClientConfiguration** cmdlet. For more information about installing Windows PowerShell, and a list of cmdlets see, How to Load the Windows PowerShell Cmdlets for App-V and Get Cmdlet Help.

2. To modify the client configuration, open a Windows PowerShell Command prompt and run **Set-AppVClientConfiguration** with any required parameters. For example:

```
$config = Get-AppVClientConfiguration
```

```
Set-AppVClientConfiguration $config
```

```
Set-AppVClientConfiguration –Name1 MyConfig –Name2 "xyz"
```

For App-V issues, use the App-V TechNet Forum.

Related topics

Operations for App-V

How to configure the client to receive package and connection groups updates from the publishing server

5/31/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

The App-V publishing server's single-point management and high scalability lets you deploy packages and connection groups and keep them up to date.

This article will tell you how to configure the App-V client to receive updates from the publishing server.

> **NOTE**
> The following example has the management server installed on a computer named **MyMgmtSrv**, and the publishing server installed on a computer named **MyPubSrv**. If the computers you'll be configuring the App-V client on have different names, you should replace the example's names with your computer's names.

Configure the App-V client to receive updates from the publishing server

1. Deploy the App-V management and publishing servers, and add the required packages and connection groups. For more information about adding packages and connection groups, see How to add or upgrade packages by using the Management Console and How to create a connection group.

2. To open the management console, open a web browser and enter the following URL: https://MyMgmtSrv/AppvManagement/Console.html. Import, publish, and entitle all packages and connection groups that your users will need.

3. On the computer running the App-V client, open an elevated Windows PowerShell command prompt, and run the following command:

   ```
   Add-AppvPublishingServer -Name ABC -URL https://MyPubSrv/AppvPublishing
   ```

   This command will configure the specified publishing server. You should see output similar to the following:

   ```
   Id                        : 1
   SetByGroupPolicy          : False
   Name                      : ABC
   URL                       : https://MyPubSrv/AppvPublishing
   GlobalRefreshEnabled      : False
   GlobalRefreshOnLogon      : False
   GlobalRefreshInterval     : 0
   GlobalRefreshIntervalUnit : Day
   UserRefreshEnabled        : True
   UserRefreshOnLogon        : True
   UserRefreshInterval       : 0
   UserRefreshIntervalUnit   : Day
   ```

4. On the computer running the App-V client, open a Windows PowerShell command prompt and enter the following cmdlet:

```
Sync-AppvPublishingServer -ServerId 1
```

   This cmdlet will query the publishing server for which packages and connection groups need to be added or removed for this particular client based on your configured entitlements for the packages and connection groups on the management server.

Related topics

- Operations for App-V

How to apply the user configuration file by using Windows PowerShell

6/6/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

When you publish a package to a specific user, you'll also need to specify a dynamic user configuration file to tell that package how to run.

Apply a user configuration file

Here's how to specify a user-specific configuration file:

> **NOTE**
>
> The following example cmdlets use this example file path for its package:
>
> - C:\Packages\Contoso\MyApp.appv.
>
> If your package file uses a different file path than the example, feel free to replace it as needed.

1. Enter the following cmdlet in Windows PowerShell to add the package to the computer:

```
Add-AppVClientPackage C:\Packages\Contoso\MyApp.appv
```

2. Enter the following cmdlet to publish the package to the user and specify the updated the dynamic user configuration file:

```
Publish-AppVClientPackage $pkg -DynamicUserConfigurationPath C:\Packages\Contoso\config.xml
```

Related topics

- Operations for App-V

How to apply the deployment configuration file by using Windows PowerShell

6/6/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

When you add or set a package to a computer running the App-V client before it's been published, a dynamic deployment configuration file is applied to it. The dynamic deployment configuration file configures the default settings for the package that all users share on the computer running the App-V client. This section will tell you how to use a deployment configuration file.

Apply the deployment configuration file with Windows PowerShell

> **NOTE**
>
> The following example cmdlet uses the following two file paths for the package and configuration files:
>
> - C:\Packages\Contoso\MyApp.appv
> - C:\Packages\Contoso\DynamicConfigurations\deploymentconfig.xml
>
> If your package and configuration files use different file paths than the example, feel free to replace them as needed.

To specify a new default set of configurations for all users who will run the package on a specific computer, in a Windows PowerShell console, enter the following cmdlet:

```
Add-AppVClientPackage -Path C:\Packages\Contoso\MyApp.appv -DynamicDeploymentConfiguration
C:\Packages\Contoso\DynamicConfigurations\deploymentconfig.xml
```

> **NOTE**
>
> This command captures the resulting object into $pkg. If the package is already present on the computer, you can use the **Set-AppVclientPackage** cmdlet to apply the deployment configuration document:
>
> ```
> Set-AppVClientPackage -Name Myapp -Path C:\Packages\Contoso\MyApp.appv -DynamicDeploymentConfiguration
> C:\Packages\Contoso\DynamicConfigurations\deploymentconfig.xml
> ```

Related topics

- Operations for App-V

How to Sequence a Package by using Windows PowerShell

6/10/2019 • 2 minutes to read • Edit Online

Applies to

- Windows 10, version 1607

Use the following procedure to create a new App-V package using Windows PowerShell.

> **NOTE**
> Before you use this procedure you must copy the associated installer files to the computer running the sequencer and you have read and understand the sequencer section of Planning for the App-V Sequencer and Client Deployment.

To create a new virtual application by using Windows PowerShell

1. Install the App-V sequencer. For more information about installing the sequencer see How to Install the Sequencer.
2. Click **Start** and type **Windows PowerShell**. Right-click **Windows PowerShell**, and select **Run as Administrator**.
3. Using the Windows PowerShell console, type the following: **import-module appvsequencer**.
4. To create a package, use the **New-AppvSequencerPackage** cmdlet. The following parameters are required to create a package:
   - **Name** - specifies the name of the package.
   - **PrimaryVirtualApplicationDirectory** - specifies the path to the directory that will be used to install the application. This path must exist.
   - **Installer** - specifies the path to the associated application installer.
   - **Path** - specifies the output directory for the package.

   For example:

   ```
   New-AppvSequencerPackage –Name <name of package> -PrimaryVirtualApplicationDirectory <path to the package root> -Installer <path to the installer executable> -OutputPath <directory of the output path>
   ```

Wait for the sequencer to create the package. Creating a package by using Windows PowerShell can take time. If the package was not created successfully, an error will be returned.

The following list displays additional optional parameters that can be used with **New-AppvSequencerPackage** cmdlet:

- AcceleratorFilePath – specifies the path to the accelerator .cab file to generate a package.
- InstalledFilesPath - specifies the path to where the local installed files of the application are saved.
- InstallMediaPath - specifies the path of the installation media

- TemplateFilePath - specifies the path to a template file if you want to customize the sequencing process.
- FullLoad - specifies that the package must be fully downloaded to the computer running the App-V before it can be opened.

In Windows 10, version 1703, running the new-appvsequencerpackage or the update-appvsequencepackage cmdlets automatically captures and stores all of your customizations as an App-V project template. If you want to make changes to this package later, your customizations are automatically loaded from this template file.

> **IMPORTANT**
>
> If you have an auto-saved template and you attempt to load another template through the *TemplateFilePath* parameter, the customization value from the parameter will override the auto-saved template.

Related topics

- Administering App-V by using Windows PowerShell

For App-V issues, use the App-V TechNet Forum.

How to create a package accelerator by using Windows PowerShell

6/6/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10, version 1607

App-V Package Accelerators automatically sequence large, complex applications. Also, when you apply an App-V Package Accelerator, you don't have to manually install an application to create the virtualized package.

Create a package accelerator

1. Install the App-V sequencer. For more information about installing the sequencer, seeHow to install the sequencer.
2. To open a Windows PowerShell console, select **Start** and enter **PowerShell**. Right-click **Windows PowerShell** and select **Run as Administrator**.
3. Make sure that you have the .appv package to create an accelerator from the installation media or installation files. You can also optionally use a readme file for the accelerator's users to reference.
4. Enter the **New-AppvPackageAccelerator** cmdlet.

   The following parameters are required to use the package accelerator cmdlet:

   - *InstalledFilesPath* specifies the application installation path.
   - *Installer* specifies the path to the application installer media.
   - *InputPackagePath* specifies the path to the .appv package.
   - *Path* specifies the output directory for the package.

     The following example cmdlet shows how you can create a package accelerator with an .appv package and the installation media:

     ```
     New-AppvPackageAccelerator -InputPackagePath <path to the .appv file> -Installer <path to the installer executable> -Path <directory of the output path>
     ```

     You can also use the following optional parameter with the **New-AppvPackageAccelerator** cmdlet:

   - *AcceleratorDescriptionFile* specifies the path to user-created package accelerator instructions. The package accelerator instructions are **.txt** or **.rtf** description files that will be included in the package created by the package accelerator.

Related topics

- Administering App-V by using Windows PowerShell

How to Enable Reporting on the App-V Client by Using Windows PowerShell

6/6/2019 • 2 minutes to read • Edit Online

Applies to

- Windows 10, version 1607

Use the following procedure to configure the App-V for reporting.

To configure the computer running the App-V client for reporting

1. Enable the App-V client. For more information, see Enable the App-V desktop client.
2. After you have enabled the App-V client, use the **Set-AppvClientConfiguration** cmdlet to configure appropriate Reporting Configuration settings:

SETTING	DESCRIPTION
ReportingEnabled	Enables the client to return information to a reporting server. This setting is required for the client to collect the reporting data on the client.
ReportingServerURL	Specifies the location on the reporting server where client information is saved. For example, https://<reportingservername>:<reportingportnumber>. **Note** This is the port number that was assigned during the Reporting Server setup
Reporting Start Time	This is set to schedule the client to automatically send the data to the server. This setting will indicate the hour at which the reporting data will start to send. It is in the 24 hour format and will take a number between 0-23.
ReportingRandomDelay	Specifies the maximum delay (in minutes) for data to be sent to the reporting server. When the scheduled task is started, the client generates a random delay between 0 and ReportingRandomDelay and will wait the specified duration before sending data.
ReportingInterval	Specifies the retry interval that the client will use to resend data to the reporting server.
ReportingDataCacheLimit	Specifies the maximum size in megabytes (MB) of the XML cache for storing reporting information. The size applies to the cache in memory. When the limit is reached, the log file will roll over.
ReportingDataBlockSize	Specifies the maximum size in megabytes (MB) of the XML cache for storing reporting information. The size applies to the cache in memory. When the limit is reached, the log file will roll over.

3. After the appropriate settings have been configured, the computer running the App-V client will automatically collect data and will send the data back to the reporting server.

   Additionally, administrators can manually send the data back in an on-demand manner using the **Send-AppvClientReport** cmdlet.

For App-V issues, use the App-V TechNet Forum.

Related topics

Administering App-V by Using Windows PowerShell

How to Install the App-V Databases and Convert the Associated Security Identifiers by Using Windows PowerShell

5/31/2019 • 3 minutes to read • Edit Online

Applies to

- Windows Server 2016

Use the following Windows PowerShell procedure to convert any number of Active Directory Domain Services (AD DS) user or machine accounts into formatted Security Identifiers (SIDs) both in the standard format and in the hexadecimal format used by Microsoft SQL Server when running SQL scripts.

Before attempting this procedure, you should read and understand the information and examples displayed in the following list:

- **.INPUTS** – The account or accounts used to convert to SID format. This can be a single account name or an array of account names.
- **.OUTPUTS** - A list of account names with the corresponding SID in standard and hexadecimal formats.
- **Examples** -

  .\ConvertToSID.ps1 DOMAIN\user_account1 DOMAIN\machine_account1$ DOMAIN\user_account2 | Format-List.

  $accountsArray = @("DOMAIN\user_account1", "DOMAIN\machine_account1$", "DOMAIN_user_account2")

  .\ConvertToSID.ps1 $accountsArray | Write-Output -FilePath .\SIDs.txt -Width 200

  #>

To convert any number of Active Directory Domain Services (AD DS) user or machine accounts into formatted Security Identifiers (SIDs)

1. Copy the following script into a text editor and save it as a Windows PowerShell script file, for example **ConvertToSIDs.ps1**.
2. To open a Windows PowerShell console, click **Start** and type **PowerShell**. Right-click **Windows PowerShell** and select **Run as Administrator**.

```
<#
.SYNOPSIS
This Windows PowerShell script will take an array of account names and try to convert each of them to
the corresponding SID in standard and hexadecimal formats.
.DESCRIPTION
This is a Windows PowerShell script that converts any number of Active Directory (AD) user or machine
accounts into formatted Security Identifiers (SIDs) both in the standard format and in the hexadecimal
format used by SQL server when running SQL scripts.
.INPUTS
The account(s) to convert to SID format. This can be a single account name or an array of account
names. Please see examples below.
.OUTPUTS
A list of account names with the corresponding SID in standard and hexadecimal formats
.EXAMPLE
```

```
.EXAMPLE
.\ConvertToSID.ps1 DOMAIN\user_account1 DOMAIN\machine_account1$ DOMAIN\user_account2 | Format-List
.EXAMPLE
$accountsArray = @("DOMAIN\user_account1", "DOMAIN\machine_account1$", "DOMAIN_user_account2")
.\ConvertToSID.ps1 $accountsArray | Write-Output -FilePath .\SIDs.txt -Width 200
#>

[]()

[]()
function ConvertSIDToHexFormat
{
   param([System.Security.Principal.SecurityIdentifier]$sidToConvert)
   $sb = New-Object System.Text.StringBuilder
    [int] $binLength = $sidToConvert.BinaryLength
    [Byte[]] $byteArray = New-Object Byte[] $binLength
   $sidToConvert.GetBinaryForm($byteArray, 0)
   foreach($byte in $byteArray)
   {
   $sb.Append($byte.ToString("X2")) |Out-Null
   }
   return $sb.ToString()
}
 [string[]]$myArgs = $args
if(($myArgs.Length -lt 1) -or ($myArgs[0].CompareTo("/?") -eq 0))
{
 [string]::Format("{0}====== Description ======{0}{0}" +
"  Converts any number of user or machine account names to string and hexadecimal SIDs.{0}" +
               "  Pass the account(s) as space separated command line parameters. (For example
'ConvertToSID.exe DOMAIN\\Account1 DOMAIN\\Account2 ...'){0}" +
               "  The output is written to the console in the format 'Account name    SID as string
SID as hexadecimal'{0}" +
               "  And can be written out to a file using standard Windows PowerShell redirection{0}" +
               "  Please specify user accounts in the format 'DOMAIN\username'{0}" +
               "  Please specify machine accounts in the format 'DOMAIN\machinename$'{0}" +
               "  For more help content, please run 'Get-Help ConvertToSID.ps1'{0}" +
               "{0}====== Arguments ======{0}" +
               "{0}  /?    Show this help message", [Environment]::NewLine)
{
else
{
    #If an array was passed in, try to split it
    if($myArgs.Length -eq 1)
    {
        $myArgs = $myArgs.Split(' ')
    }

    #Parse the arguments for account names
    foreach($accountName in $myArgs)
    {
        [string[]] $splitString = $accountName.Split('\')  # We're looking for the format
"DOMAIN\Account" so anything that does not match, we reject
        if($splitString.Length -ne 2)
        {
            $message = [string]::Format("{0} is not a valid account name. Expected format
'Domain\username' for user accounts or 'DOMAIN\machinename$' for machine accounts.", $accountName)
            Write-Error -Message $message
            continue
        }
        #Convert any account names to SIDs
        try
        {
            [System.Security.Principal.NTAccount] $account = New-Object
System.Security.Principal.NTAccount($splitString[0], $splitString[1])
            [System.Security.Principal.SecurityIdentifier] $SID =
[System.Security.Principal.SecurityIdentifier]
($account.Translate([System.Security.Principal.SecurityIdentifier]))
        }
        catch [System.Security.Principal.IdentityNotMappedException]
        {
```

```
        {
            $message = [string]::Format("Failed to translate account object '{0}' to a SID. Please
verify that this is a valid user or machine account.", $account.ToString())
            Write-Error -Message $message
            continue
        }

        #Convert regular SID to binary format used by SQL
        $hexSIDString = ConvertSIDToHexFormat $SID
        $SIDs = New-Object PSObject
        $SIDs | Add-Member NoteProperty Account $accountName
        $SIDs | Add-Member NoteProperty SID $SID.ToString()
        $SIDs | Add-Member NoteProperty Hexadecimal $hexSIDString

        Write-Output $SIDs
    }
}
```

3. Run the script you saved in step one of this procedure passing the accounts to convert as arguments.

   For example,

   .\ConvertToSID.ps1 DOMAIN\user_account1 DOMAIN\machine_account1$ DOMAIN\user_account2 | Format-List" or "$accountsArray = @("DOMAIN\user_account1", "DOMAIN\machine_account1$", "DOMAIN_user_account2")

   .\ConvertToSID.ps1 $accountsArray | Write-Output -FilePath .\SIDs.txt -Width 200"

For App-V issues, use the App-V TechNet Forum.

Related topics

Administering App-V by Using Windows PowerShell

Troubleshooting App-V

5/31/2019 • 2 minutes to read • Edit Online

Applies to

- Windows 10, version 1607

For information that can help with troubleshooting App-V for Windows 10, see:

- Application Virtualization (App-V): List of Microsoft Support Knowledge Base Articles
- Microsoft App-V Team Blog
- Release Notes for App-V
- Technical Reference for App-V
- App-V TechNet Forum

Other resources

- Application Virtualization (App-V) for Windows 10 overview
- Getting Started with App-V for Windows 10
- Planning for App-V
- Deploying App-V
- Operations for App-V

For App-V issues, use the App-V TechNet Forum.

Technical Reference for App-V

5/31/2019 • 2 minutes to read • Edit Online

Applies to

- Windows 10, version 1607

This section provides reference information related to managing App-V.

In This Section

- Performance Guidance for Application Virtualization

  Provides strategy and context for a number of performance optimization practices. Not all practices will be applicable although they are supported and have been tested. Using all suggested practices that are applicable to your organization will provide the optimal end-user experience.

- Application Publishing and Client Interaction

  Describes how the following App-V client operations affect the local operating system: App-V files and data storage locations, package registry, package store behavior, roaming registry and data, client application lifecycle management, integration of App-V packages, dynamic configuration, side-by-side assemblies, and client logging.

- Viewing App-V Server Publishing Metadata

  Tells how to view publishing metadata, which can help you resolve publishing-related issues.

- Running a Locally Installed Application Inside a Virtual Environment with Virtualized Applications

  Describes reasons and methods for running a locally installed application in a virtual environment, alongside applications that have been virtualized by using Application Virtualization (App-V).

For App-V issues, use the App-V TechNet Forum.

Related topics

How to Deploy the App-V Databases by Using SQL Scripts

Administering App-V by Using Windows PowerShell

Windows PowerShell reference for App-V

Available Mobile Device Management (MDM) settings for App-V

5/31/2019 • 2 minutes to read • Edit Online

With Windows 10, version 1703, you can configure, deploy, and manage your App-V apps with the following Mobile Device Management (MDM) settings. For the full list of available settings, see the EnterpriseAppVManagement CSP page.

POLICY NAME	SUPPORTED VERSIONS	URI FULL PATH	DATA TYPE	VALUES
Name	Windows 10, version 1703	./Vendor/MSFT/EnterpriseAppVManagement/AppVPackageManagement/<enterprise_id>/<package_family_name>/<package_full_name>/Name	String	Read-only data, provided by your App-V packages.
Version	Windows 10, version 1703	./Vendor/MSFT/EnterpriseAppVManagement/AppVPackageManagement/<enterprise_id>/<package_family_name>/<package_full_name>/Version	String	Read-only data, provided by your App-V packages.
Publisher	Windows 10, version 1703	./Vendor/MSFT/EnterpriseAppVManagement/AppVPackageManagement/<enterprise_id>/<package_family_name>/<package_full_name>/Publisher	String	Read-only data, provided by your App-V packages.
InstallLocation	Windows 10, version 1703	./Vendor/MSFT/EnterpriseAppVManagement/AppVPackageManagement/<enterprise_id>/<package_family_name>/<package_full_name>/InstallLocation	String	Read-only data, provided by your App-V packages.

POLICY NAME	SUPPORTED VERSIONS	URI FULL PATH	DATA TYPE	VALUES
InstallDate	Windows 10, version 1703	./Vendor/MSFT/EnterpriseAppVManagement/AppVPackageManagement/<enterprise_id>/<package_family_name>/<package_full_name>/InstallDate	String	Read-only data, provided by your App-V packages.
Users	Windows 10, version 1703	./Vendor/MSFT/EnterpriseAppVManagement/AppVPackageManagement/<enterprise_id>/<package_family_name>/<package_full_name>/Users	String	Read-only data, provided by your App-V packages.
AppVPackageID	Windows 10, version 1703	./Vendor/MSFT/EnterpriseAppVManagement/AppVPackageManagement/<enterprise_id>/<package_family_name>/<package_full_name>/AppVPackageID	String	Read-only data, provided by your App-V packages.
AppVVersionID	Windows 10, version 1703	./Vendor/MSFT/EnterpriseAppVManagement/AppVPackageManagement/<enterprise_id>/<package_family_name>/<package_full_name>/AppVVersionID	String	Read-only data, provided by your App-V packages.
AppVPackageUri	Windows 10, version 1703	./Vendor/MSFT/EnterpriseAppVManagement/AppVPackageManagement/<enterprise_id>/<package_family_name>/<package_full_name>/AppVPackageUri	String	Read-only data, provided by your App-V packages.
LastError	Windows 10, version 1703	./Vendor/MSFT/EnterpriseAppVManagement/AppVPublishing/LastSync/LastError	String	Read-only data, provided by your App-V packages.

POLICY NAME	SUPPORTED VERSIONS	URI FULL PATH	DATA TYPE	VALUES
LastErrorDescription	Windows 10, version 1703	./Vendor/MSFT/EnterpriseAppVManagement/AppVPublishing/LastSync/LastErrorDescription	String	- **0**: No errors returned during publish. - **1**: Unpublish groups failed during publish. - **2**: Publish no-group packages failed during publish. - **3**: Publish group packages failed during publish. - **4**: Unpublish packages failed during publish. - **5**: New policy write failed during publish. - **6**: Multiple non-fatal errors occurred during publish.
SyncStatusDescription	Windows 10, version 1703	./Vendor/MSFT/EnterpriseAppVManagement/AppVPublishing/LastSync/SyncStatusDescription	String	- **0**: App-V publishing is idle. - **1**: App-V connection groups publish in progress. - **2**: App-V packages (non-connection group) publish in progress. - **3**: App-V packages (connection group) publish in progress. - **4**: App-V packages unpublish in progress.
SyncProgress	Windows 10, version 1703	./Vendor/MSFT/EnterpriseAppVManagement/AppVPublishing/LastSync/SyncProgress	String	- **0**: App-V Sync is idle. - **1**: App-V Sync is initializing. - **2**: App-V Sync is in progress. - **3**: App-V Sync is complete. - **4**: App-V Sync requires device reboot.
PublishXML	Windows 10, version 1703	./Vendor/MSFT/EnterpriseAppVManagement/AppVPublishing/Sync/PublishXML	String	Custom value, entered by admin.
Policy	Windows 10, version 1703	./Vendor/MSFT/EnterpriseAppVManagement/AppVDynamicPolicy/configurationid/Policy	String	Custom value, entered by admin.

Performance Guidance for Application Virtualization

6/6/2019 • 20 minutes to read • Edit Online

Applies to

- Windows 7 SP1
- Windows 10
- Server 2012 R2
- Server 2016

Learn how to configure App-V for optimal performance, optimize virtual app packages, and provide a better user experience with RDS and VDI.

Implementing multiple methods can help you improve the end-user experience. However, your environment may not support all methods.

You should read and understand the following information before reading this document.

- Application Virtualization (App-V) overview
- Application Publishing and Client Interaction
- App-V Sequencing Guide

Note Some terms used in this document may have different meanings depending on external source and context. For more information about terms used in this document followed by an asterisk * review the Application Virtualization Performance Guidance Terminology section of this document.

Finally, this document will provide you with the information to configure the computer running App-V client and the environment for optimal performance. Optimize your virtual application packages for performance using the sequencer, and to understand how to use User Experience Virtualization (UE-V) or other user environment management technologies to provide the optimal user experience with App-V in both Remote Desktop Services (RDS) and non-persistent virtual desktop infrastructure (VDI).

To help determine what information is relevant to your environment you should review each section's brief overview and applicability checklist.

App-V in stateful* non-persistent deployments

This section provides information about an approach that helps ensure a user will have access to all virtual applications within seconds after logging in. This is achieved by uniquely addressing the often long-running App-V publishing refresh. As you will discover the basis of the approach, the fastest publishing refresh, is one that doesn't have to actually do anything. A number of conditions must be met and steps followed to provide the optimal user experience.

Use the information in the following section for more information:

Usage Scenarios - As you review the two scenarios, keep in mind that these are the approach extremes. Based on your usage requirements, you may choose to apply these steps to a subset of users and/or virtual applications packages.

- Optimized for Performance – To provide the optimal experience, you can expect the base image to include some of the App-V virtual application package. This and other requirements are discussed.

- Optimized for Storage – If you are concerned with the storage impact, following this scenario will help address those concerns.

Preparing your Environment

- Steps to Prepare the Base Image – Whether in a non-persistent VDI or RDSH environment, only a few steps must be completed in the base image to enable this approach.
- Use UE-V as the User Profile Management (UPM) solution for the App-V approach – the cornerstone of this approach is the ability of a UEM solution to persist the contents of just a few registry and file locations. These locations constitute the user integrations*. Be sure to review the specific requirements for the UPM solution.

User Experience Walk-through

- Walk-through – This is a step-by-step walk-through of the App-V and UE-V operations and the expectations users should have.
- Outcome – This describes the expected results.

Impact to Package Lifecycle

Enhancing the VDI Experience through Performance Optimization/Tuning

Applicability Checklist

Deployment Environment

☐	Non-Persistent VDI or RDSH.
☐	User Experience Virtualization (UE-V), other UPM solutions or User Profile Disks (UPD).

Expected Configuration

☐	User Experience Virtualization (UE-V) with the App-V user state template enabled or User Profile Management (UPM) software. Non-UE-V UPM software must be capable of triggering on Login or Process/Application Start and Logoff.
☐	App-V Shared Content Store (SCS) is configured or can be configured.

IT Administration

☐	Admin may need to update the VM base image regularly to ensure optimal performance or Admin may need to manage multiple images for different user groups.

Usage Scenarios

As you review the two scenarios, keep in mind that these approach the extremes. Based on your usage requirements, you may choose to apply these steps to a subset of users, virtual application packages, or both.

OPTIMIZED FOR PERFORMANCE	OPTIMIZED FOR STORAGE
To provide the most optimal user experience, this approach leverages the capabilities of a UPM solution and requires additional image preparation and can incur some additional image management overhead. The following describes many performance improvements in stateful non-persistent deployments. For more information, see Sequencing Steps to Optimize Packages for Publishing Performance later in this topic.	The general expectations of the previous scenario still apply here. However, keep in mind that VM images are typically stored in very costly arrays; a slight alteration has been made to the approach. Do not pre-configure user-targeted virtual application packages in the base image. The impact of this alteration is detailed in the User Experience Walk-through section of this document.

Preparing your Environment

The following table displays the required steps to prepare the base image and the UE-V or another UPM solution for the approach.

Prepare the Base Image

OPTIMIZED FOR PERFORMANCE	OPTIMIZED FOR STORAGE

• Enable the App-V client as described in Enable the App-V in-box client.
• Enable UE-V and download the App-V Settings Template from the UE-V template Gallery, see the following steps.
• Configure for Shared Content Store (SCS) mode. For more information see Deploying the App-V Sequencer and Configuring the Client.
• Configure Preserve User Integrations on Login Registry DWORD.
• Pre-configure all user- and global-targeted packages for example, **Add-AppvClientPackage**.
• Pre-configure all user- and global-targeted connection groups for example, **Add-AppvClientConnectionGroup**.
• Pre-publish all global-targeted packages.
 	
Alternatively,
 	
  ○ Perform a global publishing/refresh.
  ○ Perform a user publishing/refresh.
  ○ Un-publish all user-targeted packages.
  ○ Delete the following user-Virtual File System (VFS) entries.
 	
`AppData\Local\Microsoft\AppV\Client\VFS`
 	
`AppData\Roaming\Microsoft\AppV\Client\VFS`
	• Enable the App-V client as described in Enable the App-V in-box client.
	• Enable UE-V and download the App-V Settings Template from the UE-V template Gallery, see the following steps.
	• Configure for Shared Content Store (SCS) mode. For more information see Deploying the App-V Sequencer and Configuring the Client.
	• Configure Preserve User Integrations on Login Registry DWORD.
	• Pre-configure all global-targeted packages for example, **Add-AppvClientPackage**.
	• Pre-configure all global-targeted connection groups for example, **Add-AppvClientConnectionGroup**.
	• Pre-publish all global-targeted packages.

Configurations - For critical App-V Client configurations and for a little more context and how-to, review the following information:

CONFIGURATION SETTING	WHAT DOES THIS DO?	HOW SHOULD I USE IT?
Shared Content Store (SCS) Mode • Configurable in Windows PowerShell with `Set-AppvClientConfiguration -SharedContentStoreMode 1` or configurable with Group Policy, as described in Deploying the App-V Sequencer and Configuring the Client.	When running the shared content store only publishing data is maintained on hard disk; other virtual application assets are maintained in memory (RAM). This helps to conserve local storage and minimize disk I/O per second (IOPS).	This is recommended when low-latency connections are available between the App-V Client endpoint and the SCS content server, SAN.
PreserveUserIntegrationsOnLogin • Configure in the Registry under **HKEY_LOCAL_MACHINE** \ **Software** \ **Microsoft** \ **AppV** \ **Client** \ **Integration**. • Create the DWORD value **PreserveUserIntegrationsOnLogin** with a value of **1**. • Restart the App-V client service or restart the computer running the App-V Client.	If you have not pre-configured (**Add-AppvClientPackage**) a specific package and this setting is not configured, the App-V Client will de-integrate* the persisted user integrations, then re-integrate*. For every package that meets the above conditions, effectively twice the work will be done during publishing/refresh.	If you don't plan to pre-configure every available user package in the base image, use this setting.
MaxConcurrentPublishingRefresh • Configure in the Registry under **HKEY_LOCAL_MACHINE** \ **Software** \ **Microsoft** \ **AppV** \ **Client** \ **Publishing**. • Create the DWORD value **MaxConcurrentPublishingrefresh** with the desired maximum number of concurrent publishing refreshes. • The App-V client service and computer do not need to be restarted.	This setting determines the number of users that can perform a publishing refresh/sync at the same time. The default setting is no limit.	Limiting the number of concurrent publishing refreshes prevents excessive CPU usage that could impact computer performance. This limit is recommended in an RDS environment, where multiple users can log in to the same computer at the same time and perform a publishing refresh sync. If the concurrent publishing refresh threshold is reached, the time required to publish new applications and make them available to end users after they log in could take an indeterminate amount of time.

Configure UE-V solution for App-V Approach

We recommend using User Experience Virtualization (UE-V) to capture and centralize application settings and Windows operating system settings for a specific user. These settings are then applied to the different computers that are accessed by the user, including desktop computers, laptop computers, and virtual desktop infrastructure (VDI) sessions. UE-V is optimized for RDS and VDI scenarios.

For more information, see:

- User Experience Virtualization (UE-V) for Windows 10 overview

- Get Started with UE-V

In essence all that is required is to enable the UE-V service and download the following Microsoft authored App-V settings template from the Microsoft User Experience Virtualization (UE-V) template gallery. Register the template. For more information about UE-V templates, see User Experience Virtualization (UE-V) for Windows 10 overview.

Note Without performing an additional configuration step, User Environment Virtualization (UE-V) will not be able to synchronize the Start menu shortcuts (.lnk files) on the target computer. The .lnk file type is excluded by default.

UE-V will only support removing the .lnk file type from the exclusion list in the RDS and VDI scenarios, where every user's device will have the same set of applications installed to the same location and every .lnk file is valid for all the users' devices. For example, UE-V would not currently support the following two scenarios, because the net result will be that the shortcut will be valid on one but not all devices.

- If a user has an application installed on one device with .lnk files enabled and the same native application installed on another device to a different installation root with .lnk files enabled.
- If a user has an application installed on one device but not another with .lnk files enabled.

Important This topic describes how to change the Windows registry by using Registry Editor. If you change the Windows registry incorrectly, you can cause serious problems that might require you to reinstall Windows. You should make a backup copy of the registry files (System.dat and User.dat) before you change the registry. Microsoft cannot guarantee that the problems that might occur when you change the registry can be resolved. Change the registry at your own risk.

Using the Microsoft Registry Editor (regedit.exe), navigate to **HKEY_LOCAL_MACHINE** \ **Software** \ **Microsoft** \ **UEV** \ **Agent** \ **Configuration** \ **ExcludedFileTypes** and remove **.lnk** from the excluded file types.

Configure other User Profile Management (UPM) solutions for App-V Approach

The expectation in a stateful environment is that a UPM solution is implemented and can support persistence of user data across sessions and between logins.

The requirements for the UPM solution are as follows.

To enable an optimized login experience, for example the App-V approach for the user, the solution must be capable of:

- Persisting the below user integrations as part of the user profile/persona.
- Triggering a user profile sync on login (or application start), which can guarantee that all user integrations are applied before publishing/refresh begin, or,
- Attaching and detaching a user profile disk (UPD) or similar technology that contains the user integrations.

  Note App-V is supported when using UPD only when the entire profile is stored on the user profile disk.

  App-V packages are not supported when using UPD with selected folders stored in the user profile disk. The Copy on Write driver does not handle UPD selected folders.
- Capturing changes to the locations, which constitute the user integrations, prior to session logoff.

With App-V when you add a publishing server (**Add-AppvPublishingServer**) you can configure synchronization, for example refresh during log on and/or after a specified refresh interval. In both cases a scheduled task is created.

In previous versions of App-V, both scheduled tasks were configured using a VBScript that would initiate the user and global refresh. Starting with Hotfix Package 4 for Application Virtualization 5.0 SP2 the user refresh on log on was initiated by **SyncAppvPublishingServer.exe**. This change was introduced to provide UPM solutions a trigger process. This process delays the publish /refresh to allow the UPM solution to apply the user integrations. It will exit once the publishing/refresh is complete.

User Integrations

Registry – HKEY_CURRENT_USER

- Path - Software\Classes

  Exclude: Local Settings, ActivatableClasses, AppX*

- Path - Software\Microsoft\AppV

- Path- Software\Microsoft\Windows\CurrentVersion\App Paths

File Locations

- Root – "Environment Variable" APPDATA

  Path – Microsoft\AppV\Client\Catalog

- Root – "Environment Variable" APPDATA

  Path – Microsoft\AppV\Client\Integration

- Root – "Environment Variable" APPDATA

  Path - Microsoft\Windows\Start Menu\Programs

- (To persist all desktop shortcuts, virtual and non-virtual)

  Root - "KnownFolder" {B4BFCC3A-DB2C-424C-B029-7FE99A87C641}FileMask - *.lnk

User Experience Walk-through

This following is a step-by-step walk-through of the App-V and UPM operations and the expectations users should expect.

OPTIMIZED FOR PERFORMANCE	OPTIMIZED FOR STORAGE

After implementing this approach in the VDI/RDSH environment, on first login,
• (Operation) A user-publishing/refresh is initiated. (Expectation) If this is the first time a user has published virtual applications (e.g. non-persistent), this will take the usual duration of a publishing/refresh.
• (Operation) After the publishing/refresh, the UPM solution captures the user integrations. (Expectation) Depending on how the UPM solution is configured, this may occur as part of the logoff process. This will incur the same/similar overhead as persisting the user state.
On subsequent logins:
• (Operation) UPM solution applies the user integrations to the system prior to publishing/refresh.
(Expectation) There will be shortcuts present on the desktop, or in the start menu, which work immediately. When the publishing/refresh completes (i.e., package entitlements change), some may go away.
• (Operation) Publishing/refresh will process un-publish and publish operations for changes in user package entitlements. (Expectation) If there are no entitlement changes, publishing1 will complete in seconds. Otherwise, the publishing/refresh will increase relative to the number and complexity* of virtual applications
• (Operation) UPM solution will capture user integrations again at logoff. (Expectation) Same as previous.
[1] The publishing operation (**Publish-AppVClientPackage**) adds entries to the user catalog, maps entitlement to the user, identifies the local store, and finishes by completing any integration steps.
	After implementing this approach in the VDI/RDSH environment, on first login,
	• (Operation) A user-publishing/refresh is initiated. (Expectation)
	  ○ If this is the first time a user has published virtual applications (e.g., non-persistent), this will take the usual duration of a publishing/refresh.
	  ○ First and subsequent logins will be impacted by pre-configuring of packages (add/refresh).
	• (Operation) After the publishing/refresh, the UPM solution captures the user integrations. (Expectation) Depending on how the UPM solution is configured, this may occur as part of the logoff process. This will incur the same/similar overhead as persisting the user state
	On subsequent logins:
	• (Operation) UPM solution applies the user integrations to the system prior to publishing/refresh.
	• (Operation) Add/refresh must pre-configure all user targeted applications. (Expectation)
	  ○ This may increase the time to application availability significantly (on the order of 10's of seconds).
	  ○ This will increase the publishing refresh time relative to the number and complexity* of virtual applications.
	• (Operation) Publishing/refresh will process un-publish and publish operations for changes to user package entitlements.

OUTCOME	**OUTCOME**
• Because the user integrations are entirely preserved, there will be no work for example, integration for the publishing/refresh to complete. All virtual applications will be available within seconds of login. • The publishing/refresh will process changes to the users entitled virtual applications which impacts the experience.	Because the add/refresh must re-configure all the virtual applications to the VM, the publishing refresh time on every login will be extended.

Impact to Package Life Cycle

Upgrading a package is a crucial aspect of the package lifecycle. To help guarantee users have access to the appropriate upgraded (published) or downgraded (un-published) virtual application packages, it is recommended

you update the base image to reflect these changes. To understand why review the following section:

App-V 5.0 SP2 introduced the concept of pending states. In the past,

- If an administrator changed entitlements or created a new version of a package (upgraded) and during a publishing/refresh that package was in-use, the un-publish or publish operation, respectively, would fail.
- Now, if a package is in-use the operation will be pended. The un-publish and publish-pend operations will be processed on service restart or if another publish or un-publish command is issued. In the latter case, if the virtual application is in-use otherwise, the virtual application will remain in a pending state. For globally published packages, a restart (or service restart) often needed.

In a non-persistent environment, it is unlikely these pended operations will be processed. The pended operations, for example tasks are captured under **HKEY_CURRENT_USER \ Software \ Microsoft \ AppV \ Client \ PendingTasks**. Although this location is persisted by the UPM solution, if it is not applied to the environment prior to log on, it will not be processed.

Enhancing the VDI Experience through Performance Optimization Tuning

The following section contains lists with information about Microsoft documentation and downloads that may be useful when optimizing your environment for performance.

.NET NGEN Blog (Highly Recommended)

- How to speed up NGEN optimization

Windows Server and Server Roles

Server Performance Tuning Guidelines for

- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012
- Microsoft Windows Server 2008 R2

Server Roles

- Remote Desktop Virtualization Host
- Remote Desktop Session Host
- IIS Relevance: App-V Management, Publishing, Reporting Web Services
- File Server (SMB) Relevance: If used for App-V Content Storage and Delivery in SCS Mode

Windows Client (Guest OS) Performance Tuning Guidance

- [Microsoft Windows 7](https://download.microsoft.com/download/E/5/7/E5783D68-160B-4366-8387-114FC3E45EB4/Performance Tuning Guidelines for Windows 7 Desktop Virtualization v1.9.docx)
- Optimization Script: (Provided by Microsoft Support)
- Microsoft Windows 8
- Optimization Script: (Provided by Microsoft Support)

Sequencing Steps to Optimize Packages for Publishing Performance

Several App-V features facilitate new scenarios or enable new customer deployment scenarios. These following features can impact the performance of the publishing and launch operations.

STEP	CONSIDERATION	BENEFITS	TRADEOFFS
No Feature Block 1 (FB1, also known as Primary FB)	No FB1 means the application will launch immediately and stream fault (application requires file, DLL and must pull down over the network) during launch. If there are network limitations, FB1 will: • Reduce the number of stream faults and network bandwidth used when you launch an application for the first time. • Delay launch until the entire FB1 has been streamed.	Stream faulting decreases the launch time.	Virtual application packages with FB1 configured will need to be re-sequenced.

Removing FB1

Removing FB1 does not require the original application installer. After completing the following steps, it is suggested that you revert the computer running the sequencer to a clean snapshot.

Sequencer UI - Create a New Virtual Application Package.

1. Complete the sequencing steps up to Customize -> Streaming.
2. At the Streaming step, do not select **Optimize the package for deployment over slow or unreliable network**.
3. If desired, move on to **Target OS**.

Modify an Existing Virtual Application Package

1. Complete the sequencing steps up to Streaming.
2. Do not select **Optimize the package for deployment over a slow or unreliable network**.
3. Move to **Create Package**.

Windows PowerShell - Update an Existing Virtual Application Package.

1. Open an elevated Windows PowerShell session.
2. Import-module **appvsequencer**.
3. **Update-AppvSequencerPackage** - **AppvPackageFilePath**

   "C:\Packages\MyPackage.appv" -Installer

   "C:\PackageInstall\PackageUpgrade.exe empty.exe" -OutputPath

   "C:\UpgradedPackages"

   Note This cmdlet requires an executable (.exe) or batch file (.bat). You must provide an empty (does nothing) executable or batch file.

STEP	CONSIDERATIONS	BENEFITS	TRADEOFFS
No SXS Install at Publish (Pre-Install SxS assemblies)	Virtual Application packages do not need to be re-sequenced. SxS Assemblies can remain in the virtual application package.	The SxS Assembly dependencies will not install at publishing time.	SxS Assembly dependencies must be pre-installed.

Creating a new virtual application package on the sequencer

If, during sequencer monitoring, an SxS Assembly (such as a VC++ Runtime) is installed as part of an application's installation, SxS Assembly will be automatically detected and included in the package. The administrator will be notified and will have the option to exclude the SxS Assembly.

Client Side:

When publishing a virtual application package, the App-V Client will detect if a required SxS dependency is already installed. If the dependency is unavailable on the computer and it is included in the package, a traditional Windows Installer (**.msi**) installation of the SxS assembly will be initiated. As previously documented, simply install the dependency on the computer running the client to ensure that the Windows Installer (.msi) installation will not occur.

STEP	CONSIDERATIONS	BENEFITS	TRADEOFFS
Selectively Employ Dynamic Configuration files	The App-V client must parse and process these Dynamic Configuration files. Be conscious of size and complexity (script execution, VREG inclusions/exclusions) of the file. Numerous virtual application packages may already have User- or computer–specific dynamic configurations files.	Publishing times will improve if these files are used selectively or not at all.	Virtual application packages would need to be reconfigured individually or via the App-V server management console to remove associated Dynamic Configuration files.

Disabling a Dynamic Configuration by using Windows PowerShell

- For already published packages, you can use `Set-AppVClientPackage -Name Myapp -Path c:\Packages\Apps\MyApp.appv` without

  -DynamicDeploymentConfiguration parameter

- Similarly, when adding new packages using `Add-AppVClientPackage -Path c:\Packages\Apps\MyApp.appv`, do not use the

  -DynamicDeploymentConfiguration parameter.

For documentation on How to Apply a Dynamic Configuration, see:

- How to Apply the User Configuration File by Using Windows PowerShell
- How to Apply the Deployment Configuration File by Using Windows PowerShell

STEP	CONSIDERATIONS	BENEFITS	TRADEOFFS
Account for Synchronous Script Execution during Package Lifecycle.	If script collateral is embedded in the package, Add cmdlets may be significantly slower. Running of scripts during virtual application launch (StartVirtualEnvironment, StartProcess) and/or Add+Publish will impact the perceived performance during one or more of these lifecycle operations.	Use of Asynchronous (Non-Blocking) Scripts will ensure that the lifecycle operations complete efficiently.	This step requires working knowledge of all virtual application packages with embedded script collateral, which have associated dynamic configurations files and which reference and run scripts synchronously.
Remove Extraneous Virtual Fonts from Package.	The majority of applications investigated by the App-V product team contained a small number of fonts, typically fewer than 20.	Virtual Fonts impact publishing refresh performance.	Desired fonts will need to be enabled/installed natively. For instructions, see Install or uninstall fonts.

Determining what virtual fonts exist in the package

- Make a copy of the package.
- Rename Package_copy.appv to Package_copy.zip
- Open AppxManifest.xml and locate the following:

```
<appv:Extension Category="AppV.Fonts">
<appv:Fonts>
<appv:Font Path="[{Fonts}]\private\CalibriL.ttf" DelayLoad="true"></appv:Font>
</appv:Fonts>
```

Note If there are fonts marked as **DelayLoad**, those will not impact first launch.

Excluding virtual fonts from the package

Use the dynamic configuration file that best suits the user scope – deployment configuration for all users on computer, user configuration for specific user or users.

- Disable fonts with the deployment or user configuration.

Fonts

```
-->
<Fonts Enabled="false" />
<!--
```

App-V Performance Guidance Terminology

The following terms are used when describing concepts and actions related to App-V performance optimization.

- **Complexity** – Refers to the one or more package characteristics that may impact performance during pre-

configure (**Add-AppvClientPackage**) or integration (**Publish-AppvClientPackage**). Some example characteristics are: manifest size, number of virtual fonts, number of files.

- **De-Integrate** – Removes the user integrations
- **Re-Integrate** – Applies the user integrations.
- **Non-Persistent, Pooled** – Creates a computer running a virtual environment each time they log in.
- **Persistent, Personal** – A computer running a virtual environment that remains the same for every login.
- **Stateful** - For this document, implies that user integrations are persisted between sessions and a user environment management technology is used in conjunction with non-persistent RDSH or VDI.
- **Stateless** – Represents a scenario when no user state is persisted between sessions.
- **Trigger** – (or Native Action Triggers). UPM uses these types of triggers to initiate monitoring or synchronization operations.
- **User Experience** - In the context of App-V, the user experience, quantitatively, is the sum of the following parts:
  - From the point that users initiate a log-in to when they are able to manipulate the desktop.
  - From the point where the desktop can be interacted with to the point a publishing refresh begins (in Windows PowerShell terms, sync) when using the App-V full server infrastructure. In standalone instances, it is when the **Add-AppVClientPackage** and **Publish-AppVClientPackage** Windows PowerShell commands are initiated.
  - From start to completion of the publishing refresh. In standalone instances, this is the first to last virtual application published.
  - From the point where the virtual application is available to launch from a shortcut. Alternatively, it is from the point at which the file type association is registered and will launch a specified virtual application.
- **User Profile Management** – The controlled and structured approach to managing user components associated with the environment. For example, user profiles, preference and policy management, application control and application deployment. You can use scripting or third-party solutions configure the environment as needed.

For App-V issues, use the App-V TechNet Forum.

Related topics

Application Virtualization (App-V) overview

Application publishing and client interaction

6/26/2019 • 46 minutes to read • Edit Online

Applies to: Windows 10, version 1607

This article provides technical information about common App-V Client operations and their integration with the local operating system.

App-V package files created by the Sequencer

The Sequencer creates App-V packages and produces a virtualized application. The sequencing process creates the following files:

FILE	DESCRIPTION
.appv	- The primary package file, which contains captured assets and state information from the sequencing process. - Architecture of the package file, publishing information, and registry in a tokenized form that can be reapplied to a machine and to a specific user upon delivery.
.MSI	Executable deployment wrapper that you can use to deploy .appv files manually or by using a third-party deployment platform.
_DeploymentConfig.XML	File used to customize the default publishing parameters for all applications in a package that is deployed globally to all users on a computer that is running the App-V Client.
_UserConfig.XML	File used to customize the publishing parameters for all applications in a package that is a deployed to a specific user on a computer that is running the App-V Client.
Report.xml	Summary of messages resulting from the sequencing process, including omitted drivers, files, and registry locations.
.CAB	Optional: Package accelerator file used to automatically rebuild a previously sequenced virtual application package.
.appvt	Optional: Sequencer template file used to retain commonly reused Sequencer settings.

To learn more about sequencing, see How to Sequence a New Application with App-V.

What's in the appv file?

The appv file is a container that stores XML and non-XML files together in a single entity. This file is built from the AppX format, which is based on the Open Packaging Conventions (OPC) standard.

To view the appv file contents, make a copy of the package, and then rename the copied file to a .zip extension.

The appv file contains the following folder and files, which are used when creating and publishing a virtual

application:

NAME	TYPE	DESCRIPTION
Root	File folder	Directory that contains the file system for the virtualized application captured during sequencing.
[Content_Types].xml	XML File	List of the core content types in the appv file (for example, DLL, EXE, BIN).
AppxBlockMap.xml	XML File	Layout of the appv file, which uses File, Block, and BlockMap elements that enable location and validation of files in the App-V package.
AppxManifest.xml	XML File	Metadata for the package that contains the required information for adding, publishing, and launching the package. Includes extension points (file type associations and shortcuts) and the names and GUIDs associated with the package.
FilesystemMetadata.xml	XML File	List of the files captured during sequencing, including attributes (such as directories, files, opaque directories, empty directories, and long and short names).
PackageHistory.xml	XML File	Information about the sequencing computer (operating system version, Internet Explorer version, .Net Framework version) and process (upgrade, package version).
Registry.dat	DAT File	Registry keys and values captured during the sequencing process for the package.
StreamMap.xml	XML File	List of files for the primary and publishing feature block. The publishing feature block contains the ICO files and required portions of files (EXE and DLL) for publishing the package. When present, the primary feature block includes files that have been optimized for streaming during the sequencing process.

App-V Client data storage locations

The App-V Client performs tasks to keep virtual applications running properly and working like locally installed applications. The process of opening and running virtual applications requires mapping from the virtual file system and registry to ensure the application has the required components of a traditional application expected by users. This section describes the assets that are required to run virtual applications and lists the location where App-V stores the assets.

NAME	LOCATION	DESCRIPTION
Package Store	%ProgramData%\App-V	Default location for read-only package files.
Machine Catalog	%ProgramData%\Microsoft\AppV\Client\Catalog	Contains per-machine configuration documents.
User Catalog	%AppData%\Microsoft\AppV\Client\Catalog	Contains per-user configuration documents.
Shortcut Backups	%AppData%\Microsoft\AppV\Client\Integration\ShortCutBackups	Stores previous integration points that enable restore on package unpublish.
Copy on Write (COW) Roaming	%AppData%\Microsoft\AppV\Client\VFS	Writeable roaming location for package modification.
Copy on Write (COW) Local	%LocalAppData%\Microsoft\AppV\Client\VFS	Writeable non-roaming location for package modification.
Machine Registry	HKLM\Software\Microsoft\AppV	Contains package state information, including VReg for machine or globally published packages (Machine hive).
User Registry	HKCU\Software\Microsoft\AppV	Contains user package state information including VReg.
User Registry Classes	HKCU\Software\Classes\AppV	Contains additional user package state information.

Additional details for the table are provided in the section below and throughout the document.

Package store

The App-V Client manages the applications assets mounted in the package store. This default storage location is %ProgramData%\App-V, but you can configure it during or after setup by using the **Set-AppVClientConfiguration** Windows PowerShell cmdlet, which modifies the local registry (**PackageInstallationRoot** value under the HKLM\Software\Microsoft\AppV\Client\Streaming key). The package store must be located at a local path on the client operating system. The individual packages are stored in the package store in subdirectories named after the Package GUID and Version GUID.

The following is an example of a path to a specific application:

```
C:\ProgramData\App-V\PackGUID\VersionGUID
```

To change the default location of the package store during setup, see Enable the App-V desktop client.

Shared Content Store

If the App-V Client is configured in Shared Content Store mode, no data is written to disk when a stream fault occurs, which means that the packages require minimal local disk space (publishing data). In VDI environments where local storage can be limited, it's important to use as little disk space as possible. You can minimize disk space usage by streaming applications from a high-performance network location (such as a SAN). For more information, see Shared Content Store in Microsoft App-V 5.0 - Behind the Scenes.

> **NOTE**
> The machine and package store must be located on a local drive, even when you're using Shared Content Store configurations for the App-V Client.

Package catalogs

The App-V Client manages the following two file-based locations:

- **Catalogs (user and machine).**
- **Registry locations**—depends on how the package is targeted for publishing. There is a Catalog (data store) for the computer, and a catalog for each individual user. The Machine catalog stores global information applicable to all users or any specific user, and the User catalog stores information applicable to a specific user. The catalog is a collection of Dynamic Configurations and manifest files; there is discrete data for both file and registry per package version.

Machine catalog

The locations described in this table can be found in the %programdata%\Microsoft\AppV\Client\Catalog\ folder.

Description	Stores package documents that are available to users on the machine when packages are added and published. However, if a package is "global" at publishing time, the integrations are available to all users. If a package is non-global, the integrations are published only for specific users, but there are still global resources that are modified and visible to anyone on the client computer (such as when the package directory is in a shared disk location). If a package is available to a user on the computer (global or non-global), the manifest is stored in the Machine Catalog. When a package is published globally, there is a Dynamic Configuration file, stored in the Machine Catalog; therefore, the determination of whether a package is global is defined according to whether there is a policy file (UserDeploymentConfiguration file) in the Machine Catalog.
Default storage location	%programdata%\Microsoft\AppV\Client\Catalog This location is not the same as the Package Store location. The Package Store is the golden or pristine copy of the package files.
Files in the machine catalog	- Manifest.xml - DeploymentConfiguration.xml - UserManifest.xml (Globally Published Package) - UserDeploymentConfiguration.xml (Globally Published Package)
Additional machine catalog location, used when the package is part of a connection group	The following location is in addition to the specific package location mentioned previously as the default storage location: %programdata%\Microsoft\AppV\Client\Catalog\PackageGroups\ConGroupGUID\ConGroupVerGUID
Additional files in the machine catalog when the package is part of a connection group	- PackageGroupDescriptor.xml - UserPackageGroupDescriptor.xml (globally published Connection Group)

User catalog

The locations described in this table can be found in the appdata\roaming\Microsoft\AppV\Client\Catalog\ folder.

Description	Created during the publishing process. Contains information used for publishing the package, and for making sure that a package is provisioned to a specific user at launch. Created in a roaming location and includes user-specific publishing information. When a package is published for a user, the policy file is stored in the User Catalog. At the same time, a copy of the manifest is also stored in the User Catalog. When a package entitlement is removed for a user, the relevant package files are removed from the User Catalog. Looking at the user catalog, an administrator can view the presence of a Dynamic Configuration file, which indicates that the package is entitled for that user. For roaming users, the User Catalog needs to be in a roaming or shared location to preserve the legacy App-V behavior of targeting users by default. Entitlement and policy are tied to a user, not a computer, so they should roam with the user once they are provisioned.
Default storage location	appdata\roaming\Microsoft\AppV\Client\Catalog\Packages\PkgGUID\VerGUID
Files in the user catalog	- UserManifest.xml - DynamicConfiguration.xml or UserDeploymentConfiguration.xml
Additional user catalog location, used when the package is part of a connection group	The following location is in addition to the specific package location mentioned above: appdata\roaming\Microsoft\AppV\Client\Catalog\PackageGroups\PkgGroupGUID\PkgGroupVerGUID
Additional file in the machine catalog when the package is part of a connection group	UserPackageGroupDescriptor.xml

Shortcut backups

During the publishing process, the App-V Client backs up any shortcuts and integration points to %AppData%\Microsoft\AppV\Client\Integration\ShortCutBackups. This backup lets integration points restore to the previous versions when the package is unpublished.

Copy on Write files

The Package Store contains a pristine copy of the package files that have been streamed from the publishing server. During normal operation of an App-V application, the user or service may require changes to the files. However, these changes aren't made in the package store to preserve your ability to repair the application, which removes these changes. These locations, called Copy on Write (COW), support both roaming and non-roaming locations. The location where the modifications are stored depends where the application has been programmed to write changes to in a native experience.

COW roaming

The COW Roaming location described above stores changes to files and directories that are targeted to the typical %AppData% location or \Users\<*username*>\AppData\Roaming location. These directories and files are then roamed based on the operating system settings.

COW local

The COW Local location is similar to the roaming location, but the directories and files are not roamed to other computers, even if roaming support has been configured. The COW Local location described above stores changes applicable to typical windows and not the %AppData% location. The directories listed will vary but there

will be two locations for any typical Windows locations (for example, Common AppData and Common AppDataS). The **S** signifies the restricted location when the virtual service requests the change as a different elevated user from the signed-in users. The non-**S** location stores user-based changes.

Package registry

Before an application can access the package registry data, the App-V Client must make the package registry data available to the applications. The App-V Client uses the real registry as a backing store for all registry data.

When a new package is added to the App-V Client, a copy of the REGISTRY.DAT file from the package is created at %ProgramData%\Microsoft\AppV\Client\VREG{Version GUID}.dat. The name of the file is the version GUID with the .DAT extension. The reason this copy is made is to ensure that the actual hive file in the package is never in use, which would prevent the removal of the package at a later time.

Registry.dat from Package Store > **%ProgramData%\Microsoft\AppV\Client\Vreg\{VersionGUID}.dat**

When the first application from the package is launched on the client, the client stages or copies the contents out of the hive file, re-creating the package registry data in an alternate location under HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\AppV\Client\Packages\PackageGuid\Versions\VersionGUID\REGISTRY. The staged registry data has two distinct types of machine data and user data. Machine data is shared across all users on the machine. User data is staged for each user to a user-specific location HKCU\Software\Microsoft\AppV\Client\Packages\PackageGUID\Registry\User. The machine data is ultimately removed at package removal time, and the user data is removed on a user unpublish operation.

Package registry staging vs. connection group registry staging

When connection groups are present, the previous process of staging the registry holds true, but instead of having one hive file to process, there are more than one. The files are processed in the order in which they appear in the connection group XML, with the first writer winning any conflicts.

The staged registry persists the same way as in the single package case. Staged user registry data remains for the connection group until it is disabled; staged machine registry data is removed on connection group removal.

Virtual registry

The purpose of the virtual registry (VREG) is to provide a single merged view of the package registry and the native registry to applications. It also provides copy-on-write (COW) functionality—that is, any changes made to the registry from the context of a virtual process are made to a separate COW location. This means that the VREG must combine up to three separate registry locations into a single view based on the populated locations in the **registry COW** > **package** > **native**. When a request is made for a registry data it will locate in order until it finds the data it was requesting. Meaning if there is a value stored in a COW location it will not proceed to other locations, however, if there is no data in the COW location it will proceed to the Package and then Native location until it finds the appropriate data.

Registry locations

There are two package registry locations and two connection group locations where the App-V Client stores registry information, depending on whether the Package is published individually or as part of a connection group. There are three COW locations for packages and three for connection groups, which are created and managed by the VREG. Settings for packages and connection groups are not shared:

Single Package VReg

The registries in the following table are located in the Registry\Client\Packages\PkgGUID\ folder.

LOCATION	DESCRIPTION

LOCATION	DESCRIPTION
COW	- Machine Registry\Client\Packages\PkgGUID\REGISTRY (Only elevate process can write) - User Registry\Client\Packages\PkgGUID\REGISTRY (User Roaming anything written under HKCU except Software\Classes - User Registry Classes\Client\Packages\PkgGUID\REGISTRY (HKCU\Software\Classes writes and HKLM for non-elevated process)
Package	- Machine Registry\Client\Packages\PkgGUID\Versions\VerGuid\Registry\Machine - User Registry Classes\Client\Packages\PkgGUID\Versions\VerGUID\Registry
Native	- Native application registry location

Connection Group VReg

The registries in the following table are located in the Machine Registry\Client\PackageGroups\GrpGUID\ and User Registry Classes\Client\PackageGroups\GrpGUID\ folders.

LOCATION	DESCRIPTION
COW	- Machine Registry\Client\PackageGroups\GrpGUID\REGISTRY (only elevate process can write) - User Registry\Client\PackageGroups\GrpGUID\REGISTRY (Anything written to HKCU except Software\Classes) - User Registry Classes\Client\PackageGroups\GrpGUID\REGISTRY
Package	- Machine Registry\Client\PackageGroups\GrpGUID\Versions\VerGUID\REGISTRY - User Registry Classes\Client\PackageGroups\GrpGUID\Versions\VerGUID\REGISTRY
Native	- Native application registry location

There are two COW locations for HKLM: elevated and non-elevated processes. Elevated processes always write HKLM changes to the secure COW under HKLM. Non-elevated processes always write HKLM changes to the non-secure COW under HKCU\Software\Classes. When an application reads changes from HKLM, elevated processes will read changes from the secure COW under HKLM. Non-elevated reads from both, favoring the changes made in the unsecure COW first.

Pass-through keys

An administrator can use pass-through keys to configure certain keys to only be read from the native registry, bypassing the Package and COW locations. Pass-through locations are global to the machine (not package-specific) and can be configured by adding the path to the key, which should be treated as pass-through to the **REG_MULTI_SZ** value called **PassThroughPaths** of the key HKLM\Software\Microsoft\AppV\Subsystem\VirtualRegistry. Any key that appears under this multi-string value (and their children) will be treated as pass-through.

The following locations are configured as pass-through locations by default:

- HKEY_CURRENT_USER\SOFTWARE\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppModel
- HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppModel
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\WINEVT
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\eventlog\Application
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\WMI\Autologger
- HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib
- HKEY_LOCAL_MACHINE\SOFTWARE\Policies
- HKEY_CURRENT_USER\SOFTWARE\Policies

The purpose of pass-through keys is to ensure that a virtual application does not write registry data in the VReg that is required for non-virtual applications for successful operation or integration. The Policies key ensures that Group Policy-based settings set by the administrator are utilized and not per package settings. The AppModel key is required for integration with Windows Modern UI-based applications. Administers ideally should not modify any of the default pass-through keys, but in some instances, the admin may need to add additional pass-through keys to adjust application behavior.

App-V package store behavior

App-V manages the Package Store, which is the location where the expanded asset files from the appv file are stored. By default, this location is stored at %ProgramData%\App-V, and is limited in terms of storage capabilities only by free disk space. The package store is organized by the GUIDs for the package and version as mentioned in the previous section.

Add packages

App-V Packages are staged upon addition to the computer with the App-V Client. The App-V Client provides on-demand staging. When publishing or manually entering the **Add-AppVClientPackage** cmdlet, the data structure is built in the package store (C:\programdata\App-V\{PkgGUID}\{VerGUID}). The package files identified in the publishing block defined in the StreamMap.xml file are added to the system, and the top level folders and child files are staged to ensure proper application assets exist at launch.

Mounting packages

Packages can be explicitly loaded by entering the **Mount-AppVClientPackage** PowerShell cmdlet or by using the **App-V Client UI** to download a package. This operation completely loads the entire package into the package store.

Streaming packages

The App-V Client can be configured to change the default behavior of streaming. All streaming policies are stored under the following registry key: HKEY_LOCAL_MACHINE\Software\Microsoft\AppV\Client\Streaming. Policies are set by entering the **Set-AppvClientConfiguration** PowerShell cmdlet. The following policies apply to streaming:

POLICY	DESCRIPTION
AllowHighCostLaunch	Allows streaming over 3G and cellular networks

POLICY	DESCRIPTION
AutoLoad	Specifies the Background Load setting: 0 – Disabled 1 – Previously Used Packages only 2 – All Packages
PackageInstallationRoot	The root folder for the package store in the local machine
PackageSourceRoot	The root override where packages should be streamed from
SharedContentStoreMode	Enables the use of Shared Content Store for VDI scenarios

These settings affect the behavior of streaming App-V package assets to the client. By default, App-V only downloads the assets required after downloading the initial publishing and primary feature blocks. There are three specific behaviors in streaming packages that it's particularly important to understand:

- Background Streaming
- Optimized Streaming
- Stream Faults

Background streaming

The Windows PowerShell cmdlet **Get-AppvClientConfiguration** can be used to determine the current mode for background streaming with the AutoLoad setting and modified with either the **Set-AppvClientConfiguration** cmdlet or from the registry (HKLM\SOFTWARE\Microsoft\AppV\ClientStreaming key). Background streaming is a default setting where the Autoload setting is set to download previously used packages. The behavior based on default setting (value=1) downloads App-V data blocks in the background after the application has been launched. This setting can either be disabled altogether (value=0) or enabled for all packages (value=2), regardless of whether they have been launched.

Optimized streaming

App-V packages can be configured with a primary feature block during sequencing. This setting allows the sequencing engineer to monitor launch files for a specific application, or applications, and mark the blocks of data in the App-V package for streaming at first launch of any application in the package.

Stream faults

After the initial stream of any publishing data and the primary feature block, requests for additional files perform stream faults. These blocks of data are downloaded to the package store on an as-needed basis. This allows a user to download only a small part of the package, typically enough to launch the package and run normal tasks. All other blocks are downloaded when a user initiates an operation that requires data not currently in the package store.

Package upgrades

App-V Packages require updating throughout the lifecycle of the application. App-V Package upgrades are like the package publish operation, as each version will be created in its own PackageRoot location: %ProgramData%\App-V{PkgGUID}{newVerGUID}. The upgrade operation is optimized by creating hard links to identical and streamed files from other versions of the same package.

Package removal

The App-V Client's behavior when packages are removed depends on the package removal method. Using an App-V full infrastructure to unpublish the application, the user catalog files (machine catalog for globally published applications) are removed, but the package store location and COW locations remain. When the **Remove-AppVClientPackge** Windows PowerShell cmdlet is used to remove an App-V Package, the package store location is cleaned. Remember that unpublishing an App-V Package from the Management Server does not

perform a Remove operation. Neither operation will remove the Package Store package files.

Roaming registry and data

App-V is able to provide a near-native experience when roaming, depending on how the application being used is written. By default, App-V roams AppData that is stored in the roaming location, based on the roaming configuration of the operating system. Other locations for storage of file-based data do not roam from computer to computer, since they are in locations that are not roamed.

Roaming requirements and user catalog data storage

App-V stores data, which represents the state of the user's catalog, in the form of:

- Files under %appdata%\Microsoft\AppV\Client\Catalog
- Registry settings under HKEY_CURRENT_USER\Software\Microsoft\AppV\Client\Packages

Together, these files and registry settings represent the user's catalog, so either both must be roamed, or neither must be roamed for a given user. App-V does not support roaming %AppData%, but not roaming the user's profile (registry), or vice versa.

> **NOTE**
>
> The **Repair-AppvClientPackage** cmdlet doesn't repair the publishing state of packages where the user's App-V state under HKEY_CURRENT_USER is missing or mismatched with the data in %appdata%.

Registry-based data

App-V registry roaming falls into two scenarios, as shown in the following table.

SCENARIO	DESCRIPTION
Applications that are run as standard users	When a standard user launches an App-V application, both HKLM and HKCU for App-V applications are stored in the HKCU hive on the machine. This presents as two distinct paths: - HKLM's location is HKCU\SOFTWARE\Classes\AppV\Client\Packages\{PkgGUID}\REGISTRY\MACHINE\SOFTWARE - HKCU's location is HKCU\SOFTWARE\Microsoft\AppV\Client\Packages\{PkgGUID}\REGISTRY\USER\{UserSID}\SOFTWARE The locations are enabled for roaming based on the operating system settings.

SCENARIO	DESCRIPTION
Applications that are run with elevation	When an application is launched with elevation: - HKLM data is stored in the HKLM hive on the local computer - HKCU data is stored in the User Registry location In this scenario, these settings are not roamed with normal operating system roaming configurations, and the resulting registry keys and values are stored in the following locations: - HKLM's location is HKLM\SOFTWARE\Microsoft\AppV\Client\Packages\{PkgGUID}\{UserSID}\REGISTRY\MACHINE\SOFTWARE - HKCU's location is HKCU\SOFTWARE\Microsoft\AppV\Client\Packages\{PkgGUID}\Registry\User\{UserSID}\SOFTWARE

App-V and folder redirection

App-V supports folder redirection of the roaming AppData folder (%AppData%). When the virtual environment is started, the roaming AppData state from the user's roaming AppData directory is copied to the local cache. Conversely, when the virtual environment is shut down, the local cache that is associated with a specific user's roaming AppData is transferred to the actual location of that user's roaming AppData directory.

A typical package has several locations mapped in the user's backing store for settings in both AppData\Local and AppData\Roaming. These locations are the Copy on Write locations that are stored per user in the user's profile, and that are used to store changes made to the package VFS directories and to protect the default package VFS.

The following table shows local and roaming locations when folder redirection has not been implemented.

VFS DIRECTORY IN PACKAGE	MAPPED LOCATION OF BACKING STORE
ProgramFilesX86	C:\Users\Local\AppData\Local\Microsoft\AppV\Client\VFS\<GUID>\ProgramFilesX86
SystemX86	C:\Users\username\AppData\Local\Microsoft\AppV\Client\VFS\<GUID>\SystemX86
Windows	C:\Users\username\AppData\Local\Microsoft\AppV\Client\VFS\<GUID>\Windows
appv_ROOT	C:\Users\username\AppData\Local\Microsoft\AppV\Client\VFS\<GUID>\appv_ROOT
AppData	C:\Users\username\AppData\Local\Microsoft\AppV\Client\VFS\<GUID>\AppData

The following table shows local and roaming locations when folder redirection has been implemented for %AppData% and the location has been redirected (typically to a network location).

VFS DIRECTORY IN PACKAGE	MAPPED LOCATION OF BACKING STORE
ProgramFilesX86	C:\Users\Local\AppData\Local\Microsoft\AppV\Client\VFS\<GUID>\ProgramFilesX86

VFS DIRECTORY IN PACKAGE	MAPPED LOCATION OF BACKING STORE
SystemX86	C:\Users\Local\AppData\Local\Microsoft\AppV\Client\VFS\<GUID>\SystemX86
Windows	C:\Users\Local\AppData\Local\Microsoft\AppV\Client\VFS\<GUID>\Windows
appv_ROOT	C:\Users\Local\AppData\Local\Microsoft\AppV\Client\VFS\<GUID>\appv_ROOT
AppData	\Fileserver\users\Local\roaming\Microsoft\AppV\Client\VFS\<GUID>\AppData

The current App-V Client VFS driver can't write to network locations, so the App-V Client detects the presence of folder redirection and copies the data on the local drive during publishing and when the virtual environment starts. After the user closes the App-V application and the App-V Client closes the virtual environment, the local storage of the VFS AppData is copied back to the network, enabling roaming to additional machines, where the process will be repeated. Here's what happens during the process:

1. During publishing or virtual environment startup, the App-V Client detects the location of the AppData directory.
2. If the roaming AppData path is local or no AppData\Roaming location is mapped, nothing happens.
3. If the roaming AppData path is not local, the VFS AppData directory is mapped to the local AppData directory.

This process solves the problem of a non-local %AppData% that is not supported by the App-V Client VFS driver. However, the data stored in this new location is not roamed with folder redirection. All changes during the running of the application happen to the local AppData location and must be copied to the redirected location. The process does the following things:

1. Shuts down the App-V application, which also shuts down the virtual environment.
2. Compresses the local cache of the roaming AppData location and store it in a .zip file.
3. Uses the time stamp at the end of the .zip packaging process to name the file.
4. Records the time stamp in the HKEY_CURRENT_USER\Software\Microsoft\AppV\Client\Packages\<GUID>\AppDataTime registry as the last known AppData time stamp.
5. Calls the folder redirection process to evaluate and initiate the .zip file uploaded to the roaming AppData directory.

The time stamp is used to determine a “last writer wins” scenario if there is a conflict and is used to optimize the download of the data when the App-V application is published, or the virtual environment is started. Folder redirection will make the data available from any other clients covered by the supporting policy and will initiate the process of storing the AppData\Roaming data to the local AppData location on the client. Here's what happens during the process:

1. The user starts an application, which also starts the virtual environment.
2. The application’s virtual environment checks for the most recent time stamped .zip file, if present.
3. The virtual environment checks the registry for the last known uploaded time stamp, if present.
4. The virtual environment downloads the most recent .zip file unless the local last known upload time stamp is greater than or equal to the time stamp from the .zip file.
5. If the local last known upload time stamp is earlier than that of the most recent .zip file in the roaming AppData location, the virtual environment extracts the .zip file to the local temp directory in the user’s profile.
6. After the .zip file is successfully extracted, the local cache of the roaming AppData directory is renamed and the new data moved into place.

7. The renamed directory is deleted and the application opens with the most recently saved roaming AppData data.

This completes the successful roaming of application settings that are present in AppData\Roaming locations. The only other condition that must be addressed is a package repair operation. The process does the following things:

1. During repair, detects if the path to the user's roaming AppData directory isn't local.
2. Maps the non-local roaming AppData path targets, recreating the expected roaming and local AppData locations.
3. Deletes the time stamp stored in the registry, if present.

This process will recreate both the local and network locations for AppData and remove the registry record of the time stamp.

App-V Client application lifecycle management

In an App-V Full Infrastructure, after applications are sequenced they are managed and published to users or computers through the App-V Management and Publishing servers. This section details the operations that occur during the common App-V application lifecycle operations (Add, publishing, launch, upgrade, and removal) and the file and registry locations that are changed and modified from the App-V Client perspective. The App-V Client operations are input as PowerShell commands on the computer running the App-V Client.

This document focuses on App-V Full Infrastructure solutions. For specific information on App-V Integration with Configuration Manager 2012, see Integrating Virtual Application Management with App-V 5 and Configuration Manager 2012 SP1.

The App-V application lifecycle tasks are triggered at user sign in (default), machine startup, or as background timed operations. The settings for the App-V Client operations, including Publishing Servers, refresh intervals, package script enablement, and others, are configured (after the client is enabled) with Windows PowerShell commands. See App-V Client Configuration Settings: Windows PowerShell.

Publishing refresh

The publishing refresh process comprises several smaller operations that are performed on the App-V Client. Since App-V is an application virtualization technology and not a task scheduling technology, the Windows Task Scheduler is utilized to enable the process when the user signs in, the machine turns on, and at scheduled intervals. The client configuration during setup listed in the previous section is the preferred method when distributing the client to a large group of computers with the correct settings. These client settings can be configured with the following Windows PowerShell cmdlets:

- **Add-AppVPublishingServer** configures the client with an App-V Publishing Server that provides App-V packages.
- **Set-AppVPublishingServer** modifies the current settings for the App-V Publishing Server.
- **Set-AppVClientConfiguration** modifies the currents settings for the App-V Client.
- **Sync-AppVPublishingServer** initiates an App-V Publishing Refresh process manually. This is also utilized in the scheduled tasks created during configuration of the publishing server.

The following sections will elaborate what goes on during the publishing refresh process.

Adding an App-V package

Adding an App-V package to the client is the first step of the publishing refresh process. The end result is the same as the **Add-AppVClientPackage** cmdlet in Windows PowerShell, except the publishing refresh add process contacts the configured publishing server and passes a high-level list of applications back to the client to pull more detailed information, rather than just doing a single package add operation.

The process then configures the client for package or connection group additions or updates, then accesses the appv file. Next, the contents of the appv file are expanded and placed on the local operating system in the

appropriate locations. The following is a detailed workflow of the process, assuming the package is configured for Fault Streaming.

How to add an App-V package

1. Initiate installation manually through Windows PowerShell or Task Sequence initiation of the Publishing Refresh process.

   a. The App-V Client makes an HTTP connection and requests a list of applications based on the target. The Publishing refresh process supports targeting machines or users.

   b. The App-V Publishing Server uses the identity of the initiating target, user or machine, and queries the database for a list of entitled applications. The list of applications is provided as an XML response, which the client uses to send additional requests to the server for more information on a per-package basis.

2. The Publishing Agent on the App-V Client will evaluate any connection groups that are unpublished or disabled, since package version updates that are part of the connection group cannot be processed.

3. Configure the packages by identifying the **Add** or **Update** operations.

   a. The App-V Client utilizes the AppX API from Windows and accesses the appv file from the publishing server.

   b. The package file is opened and the **AppXManifest.xml** and **StreamMap.xml** files are downloaded to the Package Store.

   c. Completely stream publishing block data defined in the **StreamMap.xml** file. Publishing block data is stored in Package Store\PkgGUID\VerGUID\Root.

      - Icons: Targets of extension points.
      - Portable Executable Headers (PE Headers): Targets of extension points that contain the base information about the image need on disk, accessed directly or through file types.
      - Scripts: Download scripts directory for use throughout the publishing process.

   d. Populate the Package store by doing the following:

      a. Create sparse files on disk that represent the extracted package for any directories listed.

      b. Stage top-level files and directories under root.

         All other files are created when the directory is listed as sparse on disk and streamed on demand.

   e. Create the machine catalog entries. Create the **Manifest.xml** and **DeploymentConfiguration.xml** files from the package files (if no **DeploymentConfiguration.xml** file in the package a placeholder is created).

   f. Create location of the package store in the registry **HKLM\Software\Microsoft\AppV\Client\Packages\PkgGUID\Versions\VerGUID\Catalog**.

   g. Create the **Registry.dat** file from the package store to **%ProgramData%\Microsoft\AppV\Client\VReg\{VersionGUID}.dat**.

   h. Register the package with the App-V Kernel Mode Driver at **HKLM\Microsoft\Software\AppV\MAV**.

   i. Invoke scripting from the **AppxManifest.xml** or **DeploymentConfig.xml** file for Package Add timing.

4. Configure Connection Groups by adding and enabling or disabling.

5. Remove objects that are not published to the target (user or machine).

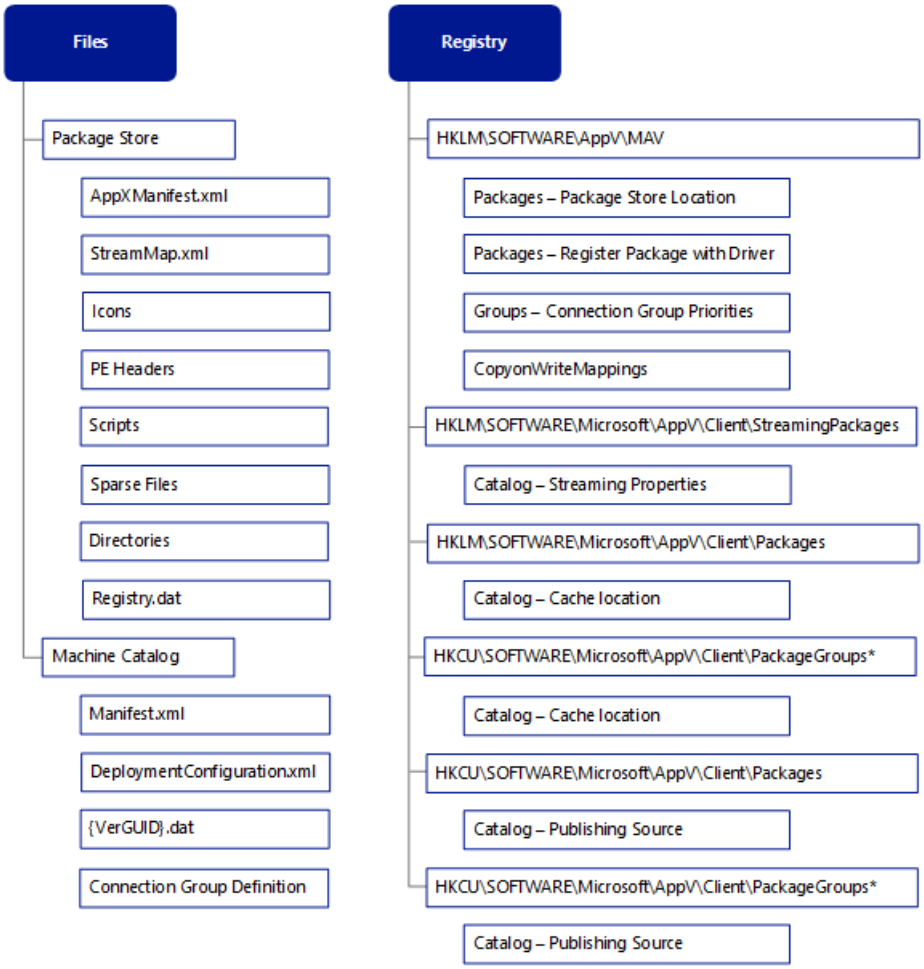
> **NOTE**
>
> This will not perform a package deletion but rather remove integration points for the specific target (user or machine) and remove user catalog files (machine catalog files for globally published).

6. Invoke background load mounting based on client configuration.

7. Packages that already have publishing information for the machine or user are immediately restored.

> **NOTE**
>
> This condition occurs as a product of removal without unpublishing with background addition of the package.

This completes an App-V package add for the publishing refresh process. The next step is publishing the package to a specific target (machine or user).



*Only present when package is in a Connection Groups

Package add file and registry data

Publishing an App-V package

During the Publishing Refresh operation, the specific publishing operation, **Publish-AppVClientPackage**, adds entries to the user catalog, maps entitlement to the user, identifies the local store, and finishes by completing any integration steps.

How to publish an App-V package

1. Package entries are added to the user catalog

a. User targeted packages: the **UserDeploymentConfiguration.xml** and **UserManifest.xml** files are placed on the machine in the User Catalog.

b. Machine targeted (global) packages: the **UserDeploymentConfiguration.xml** is placed in the Machine Catalog.

2. Register the package with the kernel mode driver for the user at **HKLM\Software\Microsoft\AppV\MAV**.

3. Perform integration tasks.

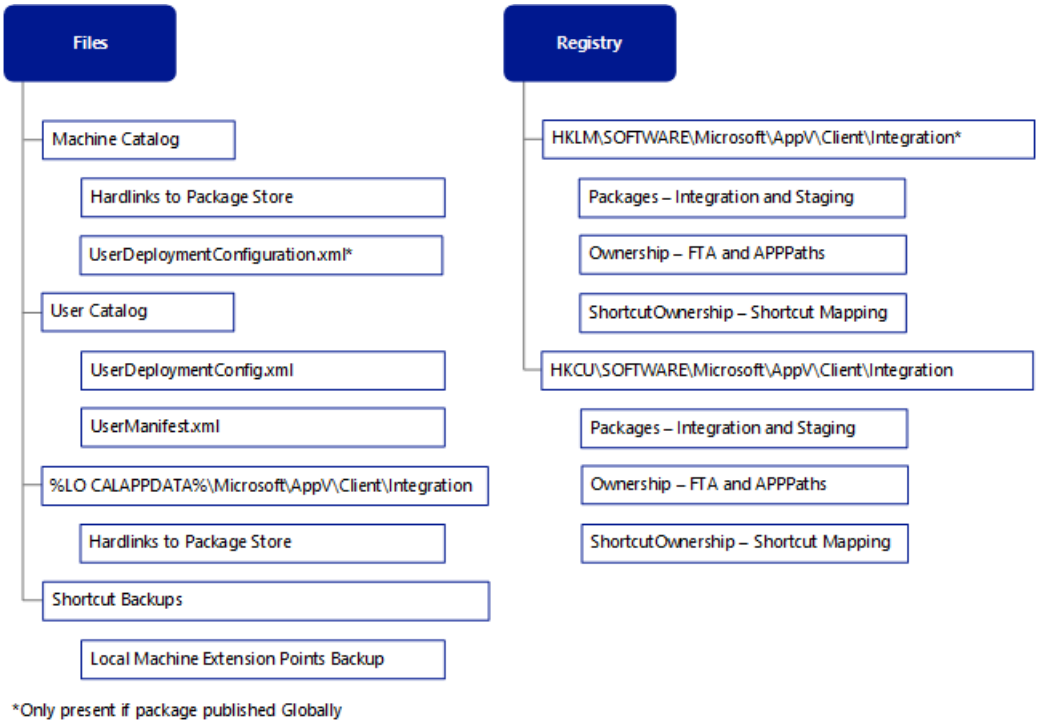
   a. Create extension points.

   b. Store backup information in the user's registry and roaming profile (Shortcut Backups).

   > **NOTE**
   >
   > This enables restore extension points if the package is unpublished.

   c. Run scripts targeted for publishing timing.

Publishing an App-V Package that is part of a Connection Group is very similar to the above process. For connection groups, the path that stores the specific catalog information includes PackageGroups as a child of the Catalog Directory. Review the Machine and User Catalog information in the preceding sections for details.



Package add file and registry data—global

Application launch

After the Publishing Refresh process, the user launches and then relaunches an App-V application. The App-V Client checks the path to the user catalog for files created during publishing. After establishing rights to launch the package, the App-V Client creates a virtual environment, begins streaming any necessary data, and applies the appropriate manifest and deployment configuration files during virtual environment creation. Once the virtual environment created and configured for the specific package and application, the application starts. This might seem like a lot, but the process in action is actually quite fast, and is optimized to minimize network traffic.

How to launch App-V applications

1. User launches the application by selecting a shortcut or file type invocation.

2. The App-V Client verifies existence in the User Catalog for the following files

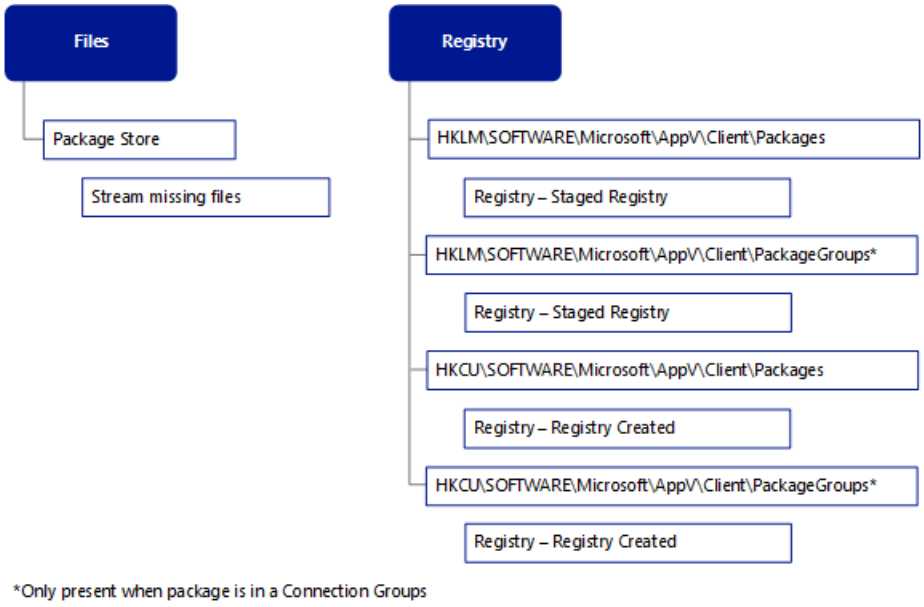
   - **UserDeploymentConfiguration.xml**
   - **UserManifest.xml**
3. If the files are present, the application is entitled for that specific user and the application will start the process for launch. There is no network traffic at this point.

4. Next, the App-V Client checks that the path for the package registered for the App-V Client service is found in the registry.

5. Upon finding the path to the package store, the virtual environment is created. If this is the first launch, the Primary Feature Block downloads if present.

6. After downloading, the App-V Client service consumes the manifest and deployment configuration files to configure the virtual environment and all App-V subsystems are loaded.

7. The Application launches. For any missing files in the package store (sparse files), App-V will stream fault the files on an as-needed basis.



Package add file and registry data—stream

Upgrading an App-V package

The current version of App-V's package upgrade process differs from the older versions in its storage optimization. App-V supports multiple versions of the same package on a machine entitled to different users. Package versions can be added at any time, as the package store and catalogs are updated with the new resources. During an upgrade in the new version, only new files are added to the new version store location, and hard links are created for unchanged files. This reduces overall storage by only presenting the file on one disk location, then projecting it into all folders with a file location entry on the disk.

How to upgrade an App-V package

1. The App-V Client performs a Publishing Refresh and discovers a newer version of an App-V Package.

2. Package entries are added to the appropriate catalog for the new version.

   a. User targeted packages: the **UserDeploymentConfiguration.xml** and **UserManifest.xml** files are placed on the machine in the user catalog at **appdata\roaming\Microsoft\AppV\Client\Catalog\Packages\PkgGUID\VerGUID**.

   b. Machine targeted (global) packages: the **UserDeploymentConfiguration.xml** is placed in the machine catalog at

%programdata%\Microsoft\AppV\Client\Catalog\Packages\PkgGUID\VerGUID.

3. Register the package with the kernel mode driver for the user at **HKLM\Software\Microsoft\AppV\MAV**.
4. Perform integration tasks.
   a. Integrate extensions points (EP) from the Manifest and Dynamic Configuration files.
   b. File based EP data is stored in the AppData folder utilizing Junction Points from the package store.
   c. Version 1 EPs already exist when a new version becomes available.
   d. The extension points are switched to the Version 2 location in machine or user catalogs for any newer or updated extension points.
5. Run scripts targeted for publishing timing.
6. Install Side-by-Side assemblies as required.

Upgrading an in-use App-V package

If you try to upgrade a package that is currently in use, the upgrade task is placed in a pending state. The upgrade will run later, according to the following rules:

TASK TYPE	APPLICABLE RULE
User-based tasks, such as publishing a package to a user	The pending task will be performed after the user logs off and then logs back on.
Globally based tasks, such as enabling a connection group globally	The pending task will be performed when the computer is shut down and then restarted.

When a task is placed in a pending state, the App-V Client also generates a registry key for the pending task, as follows:

USER-BASED OR GLOBALLY BASED TASK	WHERE THE REGISTRY KEY IS GENERATED
User-based tasks	HKEY_CURRENT_USER\Software\Microsoft\AppV\Client\PendingTasks
Globally based tasks	HKEY_LOCAL_MACHINE\Software\Microsoft\AppV\Client\PendingTasks

The following operations must be completed before users can use the newer version of the package:

TASK	DETAILS
Add the package to the computer	This task is computer-specific and you can perform it at any time by completing the steps in How to add an App-V package.
Publish the package	See the Package Publishing section above for steps. This process requires that you update extension points on the system. You can't complete this task while the application is in use.

Use the following example scenarios as a guide for updating packages.

SCENARIO	REQUIREMENTS
App-V package is not in use when you try to upgrade	None of the following components of the package can be in use: virtual application, COM server, or shell extensions. The administrator publishes a newer version of the package and the upgrade works the next time a component or application inside the package is launched. The new version of the package is streamed and ran.
App-V package is in use when the administrator publishes a newer version of the package	The App-V Client sets the operation to "pending," which means that it is queued and will be carried out later when the package is not in use. If the package application is in use, the user shuts down the virtual application, after which the upgrade can occur. If the package has shell extensions, which are permanently loaded by Windows Explorer, the user won't be able to sign in. Users must sign off and then sign back in to initiate the App-V package upgrade.

Global vs. user publishing

App-V Packages can be published in one of two ways; as user, which entitles an App-V package to a specific user or group of users, or as global, which entitles the App-V package to the entire machine for all users of the machine. Once a package upgrade has been pended and the App-V package is not in use, consider the two types of publishing:

- Global publishing is when the application is published to a machine; all users on that machine can use it. The upgrade will happen when the App-V Client Service starts, which effectively means a machine restart.
- User publishing is when the application is published to a user. If there are multiple users on the machine, the application can be published to a subset of the users. The upgrade will happen when the user signs in or when it is published again (periodically, ConfigMgr Policy refresh and evaluation, or an App-V periodic publishing/refresh, or explicitly through Windows PowerShell commands).

Removing an App-V package

Removing App-V applications in a Full Infrastructure is an unpublish operation and does not perform a package removal. The process is the same as the publish process above, but instead of adding the removal process reverses the changes that have been made for App-V Packages.

Repairing an App-V package

The repair operation is easy to do but may affect many locations on the machine. The previously mentioned Copy on Write (COW) locations are removed, and extension points are deintegrated and then reintegrated. Before repairing, please review where the COW data placement locations are registered in the registry. To perform a Repair operation, all you need to do is initiate it from the App-V Client Console or through the **Repair-AppVClientPackage** PowerShell cmdlet. After that, the operation is completed automatically.

Integration of App-V packages

The App-V Client and package architecture provides specific integration with the local operating system during the addition and publishing of packages. Three files define the integration or extension points for an App-V Package:

- AppXManifest.xml is stored inside of the package with fallback copies stored in the package store and the user profile. Contains the options created during the sequencing process.
- DeploymentConfig.xml provides configuration information of computer- and user-based integration extension

points.

- UserConfig.xml is a subset of the Deploymentconfig.xml file that only provides user-based configurations and only targets user-based extension points.

Rules of integration

When App-V applications are published to a computer with the App-V Client, some specific actions take place as described in the following list:

- Global Publishing: Shortcuts are stored in the All Users profile location and other extension points are stored in the registry in the HKLM hive.
- User Publishing: Shortcuts are stored in the current user account profile and other extension points are stored in the registry in the HKCU hive.
- Backup and Restore: Existing native application data and registry (such as FTA registrations) are backed up during publishing.
  1. App-V packages are given ownership based on the last integrated package where the ownership is passed to the newest published App-V application.
  2. Ownership transfers from one App-V package to another when the owning App-V package is unpublished. This will not initiate a restore of the data or registry.
  3. Restore the backed-up data when the last package is unpublished or removed on a per-extension point basis.

Extension points

The App-V publishing files (manifest and dynamic configuration) provide several extension points to integrate the application with the local operating system. These extension points perform typical application installation tasks, such as placing shortcuts, creating file type associations, and registering components. As these are virtualized applications that are not installed in the same manner a traditional application, there are some differences. The following is a list of extension points covered in this section:

- Shortcuts
- File type associations
- Shell extensions
- COM
- Software clients
- Application capabilities
- URL Protocol handler
- AppPath
- Virtual application

Shortcuts

The shortcut is one of the basic elements of integration with the OS and is the interface for direct user launch of an App-V application. During the publishing and unpublishing of App-V applications.

From the package manifest and dynamic configuration XML files, the path to a specific application executable can be found in a section like the following:

```
<Extension Category="AppV.Shortcut">
          <Shortcut>
            <File>[{Common Desktop}]\Adobe Reader.lnk</File>
            <Target>[{AppVPackageRoot}]\Reader\AcroRd32.exe</Target>
            <Icon>[{Windows}]\Installer\{AC76BA86-7AD7-1033-7B44-A94000000001}\SC_Reader.ico</Icon>
            <Arguments />
            <WorkingDirectory />
            <ShowCommand>1</ShowCommand>
            <ApplicationId>[{AppVPackageRoot}]\Reader\AcroRd32.exe</ApplicationId>
          </Shortcut>
        </Extension>
```

As mentioned previously, the App-V shortcuts are placed by default in the user's profile based on the refresh operation. Global refresh places shortcuts in the All Users profile and user refresh stores them in the specific user's profile. The actual executable is stored in the Package Store. The location of the ICO file is a tokenized location in the App-V package.

File type associations

Users can use file type invocations or open a file with a specifically registered extension (.docx) to start an App-V application because the App-V Client manages the local operating system File Type Associations during publishing. File type associations are present in the manifest and dynamic configuration files, as shown in the following example:

```
<Extension Category="AppV.FileTypeAssociation">
          <FileTypeAssociation>
            <FileExtension MimeAssociation="true">
              <Name>.xdp</Name>
              <ProgId>AcroExch.XDPDoc</ProgId>
              <ContentType>application/vnd.adobe.xdp+xml</ContentType>
            </FileExtension>
            <ProgId>
              <Name>AcroExch.XDPDoc</Name>
              <Description>Adobe Acrobat XML Data Package File</Description>
              <EditFlags>65536</EditFlags>
              <DefaultIcon>[{Windows}]\Installer\{AC76BA86-7AD7-1033-7B44-
A94000000001}\XDPFile_8.ico</DefaultIcon>
              <ShellCommands>
                <DefaultCommand>Read</DefaultCommand>
                <ShellCommand>
                  <ApplicationId>[{AppVPackageRoot}]\Reader\AcroRd32.exe</ApplicationId>
                  <Name>Open</Name>
                  <CommandLine>"[{AppVPackageRoot}]\Reader\AcroRd32.exe" "%1"</CommandLine>
                </ShellCommand>
                <ShellCommand>
                  <ApplicationId>[{AppVPackageRoot}]\Reader\AcroRd32.exe</ApplicationId>
                  <Name>Printto</Name>
                  <CommandLine>"[{AppVPackageRoot}]\Reader\AcroRd32.exe"  /t "%1" "%2" "%3" "%4"</CommandLine>
                </ShellCommand>
                <ShellCommand>
                  <ApplicationId>[{AppVPackageRoot}]\Reader\AcroRd32.exe</ApplicationId>
                  <Name>Read</Name>
                  <FriendlyName>Open with Adobe Reader</FriendlyName>
                  <CommandLine>"[{AppVPackageRoot}]\Reader\AcroRd32.exe" "%1"</CommandLine>
                </ShellCommand>
              </ShellCommands>
            </ProgId>
          </FileTypeAssociation>
        </Extension>
```

> **NOTE**
>
> In this example:
>
> - `<Name>.xdp</Name>` is the extension
> - `<Name>AcroExch.XDPDoc</Name>` is the ProgId value (which points to the adjoining ProgId)
> - `<CommandLine>"[{AppVPackageRoot}]\Reader\AcroRd32.exe" "%1"</CommandLine>` is the command line, which points to the application executable

Shell extensions

Shell extensions are embedded in the package automatically during the sequencing process. When the package is published globally, the shell extension gives users the same functionality as if the application were locally installed. The application requires no additional setup or configuration on the client to enable the shell extension functionality.

Requirements for using shell extensions

- Packages that contain embedded shell extensions must be published globally.
- The "bitness" of the application, Sequencer, and App-V Client must match, or the shell extensions won't work. The following example configuration fulfills the matching requirement:
  - The version of the application is 64-bit.
  - The Sequencer is running on a 64-bit computer.
  - The package is being delivered to a 64-bit App-V Client computer.

The following table displays the supported shell extensions.

HANDLER	DESCRIPTION
Context menu handler	Adds menu items to the context menu. It is called before the context menu is displayed.
Drag-and-drop handler	Controls the action upon right-click drag-and-drop and modifies the context menu that appears.
Drop target handler	Controls the action after a data object is dragged-and-dropped over a drop target, such as a file.
Data object handler	Controls the action after a file is copied to the clipboard or dragged-and-dropped over a drop target. It can provide additional clipboard formats to the drop target.
Property sheet handler	Replaces or adds pages to the property sheet dialog box of an object.
Infotip handler	Allows retrieving flags and infotip information for an item and displaying it inside a popup tooltip upon mouse-hover.
Column handler	Allows creating and displaying custom columns in Windows Explorer *Details view*. It can be used to extend sorting and grouping.
Preview handler	Enables a preview of a file to be displayed in the Windows Explorer Preview Pane.

COM

The App-V Client supports publishing applications with support for COM integration and virtualization. COM integration allows the App-V Client to register COM objects on the local operating system and virtualization of the objects. For the purposes of this document, the integration of COM objects requires additional detail.

App-V supports registering COM objects from the package to the local operating system with two process types: Out-of-process and In-process. Registering COM objects is accomplished with one or a combination of multiple modes of operation for a specific App-V package that includes Off, Isolated, and Integrated. Integrated mode is configured for either the Out-of-process or In-process type. Configuration of COM modes and types is accomplished with dynamic configuration files (deploymentconfig.xml or userconfig.xml).

For details on App-V integration, see Microsoft Application Virtualization 5.0 Integration.

Software clients and application capabilities

App-V supports specific software clients and application capabilities extension points to register virtualized applications with the operating system's software client. This means users can select default programs for operations like email, instant messaging, and using the media player. This operation is performed in the control panel with **Set Program Access** and **Computer Defaults**, and is configured during sequencing in the manifest or dynamic configuration files. Application capabilities are only supported when the App-V applications are published globally.

The following is an example of software client registration of an App-V-based mail client.

```
    <SoftwareClients Enabled="true">
      <ClientConfiguration EmailEnabled="true" />
      <Extensions>
        <Extension Category="AppV.SoftwareClient">
          <SoftwareClients>
            <EMail MakeDefault="true">
              <Name>Mozilla Thunderbird</Name>
              <Description>Mozilla Thunderbird</Description>
              <DefaultIcon>[{ProgramFilesX86}]\Mozilla Thunderbird\thunderbird.exe,0</DefaultIcon>
              <InstallationInformation>
                <RegistrationCommands>
                  <Reinstall>"[{ProgramFilesX86}]\Mozilla Thunderbird\uninstall\helper.exe"
/SetAsDefaultAppGlobal</Reinstall>
                  <HideIcons>"[{ProgramFilesX86}]\Mozilla Thunderbird\uninstall\helper.exe"
/HideShortcuts</HideIcons>
                  <ShowIcons>"[{ProgramFilesX86}]\Mozilla Thunderbird\uninstall\helper.exe"
/ShowShortcuts</ShowIcons>
                </RegistrationCommands>
                <IconsVisible>1</IconsVisible>
                <OEMSettings />
              </InstallationInformation>
              <ShellCommands>
                <ApplicationId>[{ProgramFilesX86}]\Mozilla Thunderbird\thunderbird.exe</ApplicationId>
                <Open>"[{ProgramFilesX86}]\Mozilla Thunderbird\thunderbird.exe" -mail</Open>
              </ShellCommands>
              <MAPILibrary>[{ProgramFilesX86}]\Mozilla Thunderbird\mozMapi32_InUse.dll</MAPILibrary>
              <MailToProtocol>
                <Description>Thunderbird URL</Description>
                <EditFlags>2</EditFlags>
                <DefaultIcon>[{ProgramFilesX86}]\Mozilla Thunderbird\thunderbird.exe,0</DefaultIcon>
                <ShellCommands>
                  <ApplicationId>[{ProgramFilesX86}]\Mozilla Thunderbird\thunderbird.exe</ApplicationId>
                  <Open>"[{ProgramFilesX86}]\Mozilla Thunderbird\thunderbird.exe" -osint -compose "%1"</Open>
                </ShellCommands>
              </MailToProtocol>
            </EMail>
          </SoftwareClients>
        </Extension>
      </Extensions>
    </SoftwareClients>
```

> **NOTE**
>
> In this example:
>
> - `<ClientConfiguration EmailEnabled="true" />` is the overall Software Clients setting to integrate Email clients.
> - `<EMail MakeDefault="true">` is the flag to set a particular Email client as the default Email client.
> - `<MAPILibrary>[{ProgramFilesX86}]\Mozilla Thunderbird\mozMapi32_InUse.dll</MAPILibrary>` is the MAPI dll registration.

URL Protocol handler

Virtual applications don't always specifically utilize file type invocation. For, example, in an application that supports embedding a mailto: link inside a document or web page, the user selects the link expecting to access their registered mail client. App-V supports URL Protocol handlers that can be registered on a per-package basis with the local operating system. During sequencing, the URL Protocol handlers are automatically added to the package.

For situations where there is more than one application that could register the specific URL Protocol handler, the dynamic configuration files can be utilized to modify the behavior and suppress or disable this feature for an application that should not be the primary application launched.

AppPath

The AppPath extension point supports calling App-V applications directly from the operating system. Administrators can provide access to App-V applications from operating system commands or scripts without calling the specific path to the executable from either the Run or Start Screen, depending on the operating system. It therefore avoids modifying the system path environment variable on all systems, as it is accomplished during publishing.

The AppPath extension point is configured either in the manifest or in the dynamic configuration files and is stored in the registry on the local machine during publishing for the user. For additional information on AppPath review: App Paths - A Virtual Application Extension in App-V 5.0.

Virtual application

This subsystem provides a list of applications captured during sequencing which is usually consumed by other App-V components. Integration of extension points belonging to a specific application can be disabled using dynamic configuration files. For example, if a package contains two applications, you can disable all extension points belonging to one application to only allow integration of extension points for the other application.

Extension point rules

The previously described extension points are integrated into the operating system based on how the packages has been published. Global publishing places extension points in public machine locations, where user publishing places extension points in user locations. For example, a shortcut created on the desktop and published globally will result in the file data for the shortcut (%Public%\Desktop) and the registry data (HKLM\Software\Classes). The same shortcut would have file data (%UserProfile%\Desktop) and registry data (HKCU\Software\Classes).

Extension points are not all published the same way, where some extension points will require global publishing and others require sequencing on the specific operating system and architecture where they are delivered. Below is a table that describes these two key rules.

VIRTUAL EXTENSION	REQUIRES TARGET OS SEQUENCING	REQUIRES GLOBAL PUBLISHING
Shortcut		
File Type Association		

VIRTUAL EXTENSION	REQUIRES TARGET OS SEQUENCING	REQUIRES GLOBAL PUBLISHING
URL Protocols	X	
AppPaths	X	
COM Mode		
Software Client	X	
Application Capabilities	X	X
Context Menu Handler	X	X
Drag-and-drop Handler	X	
Data Object Handler	X	
Property Sheet Handler	X	
Infotip Handler	X	
Column Handler	X	
Shell Extensions	X	
Browser Helper Object	X	X
Active X Object	X	X

Dynamic configuration processing

Deploying App-V packages to a single machine or user is very simple. However, as organizations deploy App-V applications across business lines and geographic and political boundaries, it becomes impossible to sequence all applications with the same settings. App-V was designed to overcome this problem by capturing specific settings and configurations during sequencing in the Manifest file while also supporting modification with Dynamic Configuration files.

App-V dynamic configuration lets you specify a package policy at either the machine or user levels. Sequencing engineers can use Dynamic Configuration files to modify the configuration of a package post-sequencing to address the needs of individual groups of users or machines. In some instances, it may be necessary to modify the application to provide proper functionality within the App-V environment. For example, you may need to modify the _*config.xml files to allow certain actions to be performed at a specified time while executing the application, like disabling a mailto extension to prevent a virtualized application from overwriting that extension from another application.

App-V packages contain the Manifest file inside of the App-V Package file, which is representative of sequencing operations and is the policy of choice unless Dynamic Configuration files are assigned to a specific package. Post-sequencing, the Dynamic Configuration files can be modified to allow an application to be published to different desktops or users with different extension points. The two Dynamic Configuration files are the Dynamic Deployment Configuration (DDC) and Dynamic User Configuration (DUC) files. This section focuses on the combination of the manifest and dynamic configuration files.

Examples of dynamic configuration files

The following example shows the combination of the Manifest, Deployment Configuration, and User Configuration files after publishing and during normal operation. These examples are abbreviated examples of each of the files. The purpose is show the combination of the files only, not to be a complete description of the specific categories available in each file. For more information, download the App-V Sequencing Guide.

Manifest

```
<appv:Extension Category="AppV.Shortcut">
     <appv:Shortcut>
          <appv:File>[{Common Programs}]\7-Zip\7-Zip File Manager.lnk</appv:File>
          <appv:Target>[{AppVPackageRoot}]\7zFM.exe</appv:Target>
          <appv:Icon>[{AppVPackageRoot}]\7zFM exe.0.ico</appv:Icon>
     </appv:Shortcut>
</appv:Extension>
```

Deployment Configuration

```
<MachineConfiguration>
     <Subsystems>
          <Registry>
               <Include>
                    <Key Path= "\REGISTRY\Machine\Software\7zip">
                    <Value Type="REG_SZ" Name="Config" Data="1234"/>
                    </Key>
               </Include>
          </Registry>
     </Subsystems>
```

User Configuration

```
<UserConfiguration>
     <Subsystems>
<appv:ExtensionCategory="AppV.Shortcut">
     <appv:Shortcut>
          <appv:File>[{Desktop}]\7-Zip\7-Zip File Manager.lnk</appv:File>
          <appv:Target>[{AppVPackageRoot}]\7zFM.exe</appv:Target>
          <appv:Icon>[{AppVPackageRoot}]\7zFM exe.O.ico</appv:Icon>
     </appv:Shortcut>
</appv:Extension>
     </Subsystems>
<UserConfiguration>
     <Subsystems>
<appv:Extension Category="AppV.Shortcut">
     <appv:Shortcut>
          <appv:File>[{Desktop}]\7-Zip\7-Zip File Manager.lnk</appv:File>
          <appv:Target>[{AppVPackageRoot}]\7zFM.exe</appv:Target>
          <appv:Icon>[{AppVPackageRoot}]\7zFM.exe.O.ico</appv:Icon>
     </appv:Shortcut>
     <appv:Shortcut>
          <appv:File>[{Common Programs}]\7-Zip\7-Zip File Manager.Ink</appv:File>
          <appv:Target>[{AppVPackageRoot}]\7zFM.exe</appv:Target>
          <appv:Icon>[{AppVPackageRoot)]\7zFM.exe.O.ico</appv: Icon>
     </appv:Shortcut>
</appv:Extension>
     </Subsystems>
<MachineConfiguration>
     <Subsystems>
          <Registry>
               <Include>
                    <Key Path="\REGISTRY\Machine\Software\7zip">
                    <Value Type=”REG_SZ" Name="Config" Data="1234"/>
               </Include>
          </Registry>
     </Subsystems>
```

Side-by-side assemblies

App-V supports automatic packaging of side-by-side assemblies during sequencing and deployment on the client during virtual application publishing. App-V also supports capturing side-by-side assemblies during sequencing for assemblies not present on the sequencing machine. For assemblies consisting of Visual C++ (Version 8 and newer) or MSXML run-time, the Sequencer will automatically detect and capture these dependencies even if they weren't installed during monitoring.

The side-by-side assemblies feature removes the limitations of previous versions of App-V, where the App-V Sequencer did not capture assemblies already present on the sequencing workstation, and privatized the assemblies, which limited it to one bit version per package. This behavior resulted in App-V applications being deployed to clients missing the required side-by-side assemblies, which led to application launch failures. This forced the packaging process to document and ensure that all assemblies required for packages were locally installed on the user’s client operating system. This task was both a management and implementation challenge due to the number of assemblies and the lack of application documentation for the required dependencies.

Side-by-side assembly support in App-V has the following features:

- Automatic captures of side-by-side assembly during sequencing, regardless of whether the assembly was already installed on the sequencing workstation.
- The App-V Client automatically installs required side-by-side assemblies to the client computer at publishing time if they aren't already installed.
- The Sequencer reports the VC run-time dependency in Sequencer reporting mechanism.
- The Sequencer allows opting to not package assemblies already installed on the Sequencer, supporting scenarios where the assemblies have previously been installed on the target computers.

Automatic publishing of side-by-side assemblies

During publishing of an App-V package with side-by-side assemblies, the App-V Client will check for the presence of the assembly on the machine. If it doesn't detect an assembly, the client will deploy the assembly to the machine. Packages that are part of connection groups will rely on the side-by-side assembly installations in the base packages, as the connection groups don't contain any information about assembly installation.

> **NOTE**
> Unpublishing or removing a package with an assembly does not remove the assemblies for that package.

Client logging

The App-V Client logs information to the Windows Event log in standard ETW format. The specific App-V events can be found in the event viewer under **Applications and Services Logs\Microsoft\AppV\Client**.

There are three specific categories of events recorded:

- **Admin** logs events for configurations applied to the App-V Client and also contains the primary warnings and errors.
- **Operational** logs the general App-V execution and usage of individual components, creating an audit log of the App-V Client's completed App-V operations.
- **Virtual Application** logs virtual application launches and use of virtualization subsystems.

Viewing App-V Server Publishing Metadata

5/31/2019 • 2 minutes to read • Edit Online

Applies to

- Windows Server 2016

Use this procedure to view App-V Server publishing metadata, which can help you resolve publishing-related issues. You must be using the App-V Management server to use this procedure.

This article contains the following information:

- Definition of publishing metadata
- Syntax to use for viewing publishing metadata
- Query values for client operating system

Definition of publishing metadata

When packages are published to a computer that is running the App-V client, metadata is sent to that computer indicating which packages and connection groups are being published. The App-V Client makes two separate requests for the following:

- Packages and connection groups that are entitled to the client computer.
- Packages and connection groups that are entitled to the current user.

The Publishing server communicates with the Management server to determine which packages and connection groups are available to the requester. The Publishing server must be registered with the Management server in order for the metadata to be generated.

You can view the metadata for each request in an Internet browser by using a query that is in the context of the specific user or computer.

Query syntax for viewing publishing metadata

This section provides information about queries for viewing publishing metadata for App-V 5.0 SP3 Server and App-V 5.1 server. The App-V server components have not changed since App-V 5.0 was released, so App-V 5.x Server is the version of the server used with App-V for Windows 10.

Query syntax

```
http://<PubServer>:<Publishing Port#>/?ClientVersion=<BuildNumber>&ClientOS=<OSStringValue>
```

For information about the variables in this syntax, see the table that follows.

Query example

```
http://pubsvr01:2718/?ClientVersion=10.0.14393&ClientOS=WindowsClient_10.0_x64
```

In this example:

- A computer running Windows Server 2016 named “pubsvr01” hosts the Publishing service.
- The Windows client is Windows 10, 64-bit.

Query parameter descriptions

The following table describes the parameters shown in the preceding **Query syntax**.

PARAMETER	DESCRIPTION
`<PubServer>`	Name of the App-V Publishing server.
`<Publishing Port#>`	Port to the App-V Publishing server, which you defined when you configured the Publishing server.
`ClientVersion=<BuildNumber>`	Windows 10 build number. You can obtain this number by running the following Windows PowerShell command: `(Get-CimInstance Win32_OperatingSystem).version`
`ClientOS=<OSStringValue>`	Operating system of the computer that is running the App-V client. Refer to the table that follows for the correct value. You can omit this parameter, with the result that only the packages that were sequenced to support all operating systems will appear in the metadata.

To get the name of the Publishing server and the port number (`http://<PubServer>:<Publishing Port#>`) from the App-V client, look at the URL configuration of the **Get-AppvPublishingServer** Windows PowerShell cmdlet.

Query values for client operating system

In your publishing metadata query, enter the string values that correspond to the client operating system that you're using.

OPERATING SYSTEM	ARCHITECTURE	STRING VALUE
Windows 10	64-bit	WindowsClient_10.0_x64
Windows 10	32-bit	WindowsClient_10.0_x86
Windows 8.1	64-bit	WindowsClient_6.2_x64
Windows 8.1	32-bit	WindowsClient_6.2_x86
Windows 8	64-bit	WindowsClient_6.2_x64
Windows 8	32-bit	WindowsClient_6.2_x86
Windows Server 2012 R2	64-bit	WindowsServer_6.2_x64
Windows Server 2012 R2	32-bit	WindowsServer_6.2_x86
Windows Server 2012	64-bit	WindowsServer_6.2_x64

OPERATING SYSTEM	ARCHITECTURE	STRING VALUE
Windows Server 2012	32-bit	WindowsServer_6.2_x86
Windows Server 2008 R2	64-bit	WindowsServer_6.1_x64
Windows Server 2008 R2	32-bit	WindowsServer_6.1_x86

For App-V issues, use the App-V TechNet Forum.

Related topics

Technical Reference for App-V

Running a Locally Installed Application Inside a Virtual Environment with Virtualized Applications

5/31/2019 • 4 minutes to read • Edit Online

Applies to

- Windows 7 SP1
- Windows 10
- Windows Server 2012 R2
- Windows Server 2016

You can run a locally installed application in a virtual environment, alongside applications that have been virtualized by using Microsoft Application Virtualization (App-V). You might want to do this if you:

- Want to install and run an application locally on client computers, but want to virtualize and run specific plug-ins that work with that local application.
- Are troubleshooting an App-V client package and want to open a local application within the App-V virtual environment.

Use any of the following methods to open a local application inside the App-V virtual environment:

- RunVirtual registry key
- Get-AppvClientPackage Windows PowerShell cmdlet
- Command line switch /appvpid:<PID>
- Command line hook switch /appvve:<GUID>

Each method accomplishes essentially the same task, but some methods may be better suited for some applications than others, depending on whether the virtualized application is already running.

RunVirtual registry key

To add a locally installed application to a package or to a connection group's virtual environment, you add a subkey to the `RunVirtual` registry key in the Registry Editor, as described in the following sections.

There is no Group Policy setting available to manage this registry key, so you have to use System Center Configuration Manager or another electronic software distribution (ESD) system, or manually edit the registry.

Starting with App-V 5.0 SP3, when using RunVirtual, you can publish packages globally or to the user.

Steps to create the subkey

1. Using the information in the following table, create a new registry key using the name of the executable file, for example, **MyApp.exe**.

PACKAGE PUBLISHING METHOD	WHERE TO CREATE THE REGISTRY KEY

PACKAGE PUBLISHING METHOD	WHERE TO CREATE THE REGISTRY KEY
Published globally	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\AppV\Client\RunVirtual **Example**: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\AppV\Client\RunVirtual\MyApp.exe
Published to the user	HKEY_CURRENT_USER\SOFTWARE\Microsoft\AppV\Client\RunVirtual **Example**: HKEY_CURRENT_USER\SOFTWARE\Microsoft\AppV\Client\RunVirtual\MyApp.exe
Connection group can contain: • Packages that are published just globally or just to the user • Packages that are published globally and to the user	Either HKEY_LOCAL_MACHINE or HKEY_CURRENT_USER key, but all of the following must be true: • If you want to include multiple packages in the virtual environment, you must include them in an enabled connection group. • Create only one subkey for one of the packages in the connection group. If, for example, you have one package that is published globally, and another package that is published to the user, you create a subkey for either of these packages, but not both. Although you create a subkey for only one of the packages, all of the packages in the connection group, plus the local application, will be available in the virtual environment. • The key under which you create the subkey must match the publishing method you used for the package. For example, if you published the package to the user, you must create the subkey under `HKEY_CURRENT_USER\SOFTWARE\Microsoft\AppV\Client\RunVirtual`. Do not add a key for the same application under both hives.

2. Set the new registry subkey's value to the PackageId and VersionId of the package, separating the values with an underscore.

   Syntax: <PackageId>_<VersionId>

   Example: 4c909996-afc9-4352-b606-0b74542a09c1_be463724-Oct1-48f1-8604-c4bd7ca92fa

   The application in the previous example would produce a registry export file (.reg file) like the following:

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\AppV\Client\RunVirtual]
@=""
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\AppV\Client\RunVirtual\MyApp.exe]
@="aaaaaaaa-bbbb-cccc-dddd-eeeeeeee_11111111-2222-3333-4444-555555555
```

Get-AppvClientPackage Windows PowerShell cmdlet

You can use the **Start-AppVVirtualProcess** cmdlet to retrieve the package name and then start a process within the specified package's virtual environment. This method lets you launch any command within the context of an App-V package, regardless of whether the package is currently running.

Use the following example syntax, and substitute the name of your package for **<Package>**:

```
$AppVName = Get-AppvClientPackage <Package>
```

```
Start-AppvVirtualProcess -AppvClientObject $AppVName cmd.exe
```

If you don't know the exact name of your package, you can use the command line **Get-AppvClientPackage *executable***, where **executable** is the name of the application, for example:
Get-AppvClientPackage *Word*

Command line switch /appvpid:<PID>

You can apply the **/appvpid:<PID>** switch to any command, which enables that command to run within a virtual process that you select by specifying its process ID (PID). Using this method launches the new executable in the same App-V environment as an executable that is already running.

Example: `cmd.exe /appvpid:8108`

To find the process ID (PID) of your App-V process, run the command **tasklist.exe** from an elevated command prompt.

Command line hook switch /appvve:<GUID>

This switch lets you run a local command within the virtual environment of an App-V package. Unlike the **/appvid** switch, where the virtual environment must already be running, this switch enables you to start the virtual environment.

Syntax: `cmd.exe /appvve:<PACKAGEGUID_VERSIONGUID>`

Example: `cmd.exe /appvve:aaaaaaaa-bbbb-cccc-dddd-eeeeeeee_11111111-2222-3333-4444-55555555`

To get the package GUID and version GUID of your application, run the **Get-AppvClientPackage** cmdlet. Concatenate the **/appvve** switch with the following:

- A colon
- Package GUID of the desired package
- An underscore
- Version ID of the desired package

If you don't know the exact name of your package, use the command line **Get-AppvClientPackage *executable***, where **executable** is the name of the application, for example:
Get-AppvClientPackage *Word*

This method lets you launch any command within the context of an App-V package, regardless of whether the package is currently running.

For App-V issues, use the App-V TechNet Forum.

Related topics

Changes to Service Host grouping in Windows 10

5/31/2019 • 3 minutes to read • Edit Online

Applies to: Windows 10

The **Service Host (svchost.exe)** is a shared-service process that serves as a shell for loading services from DLL files. Services are organized into related host groups, and each group runs inside a different instance of the Service Host process. In this way, a problem in one instance does not affect other instances. Service Host groups are determined by combining the services with matching security requirements. For example:

- Local Service
- Local Service No Network
- Local Service Network Restricted
- Local System
- Local System Network Restricted
- Network Service

Separating SvcHost services

Beginning with Windows 10 Creators Update (version 1703), services that were previously grouped will instead be separated - each will run in its own SvcHost process. This change is automatic for systems with **more than 3.5 GB** of RAM running the Client Desktop SKU. On systems with 3.5 GB or less RAM, we'll continue to group services into a shared SvcHost process.

Benefits of this design change include:

- Increased reliability by insulating critical network services from the failure of another non-network service in the host, and adding the ability to restore networking connectivity seamlessly when networking components crash.
- Reduced support costs by eliminating the troubleshooting overhead associated with isolating misbehaving services in the shared host.
- Increased security by providing additional inter-service isolation
- Increased scalability by allowing per-service settings and privileges
- Improved resource management through per-service CPU, I/O and memory management and increase clear diagnostic data (report CPU, I/O and network usage per service).

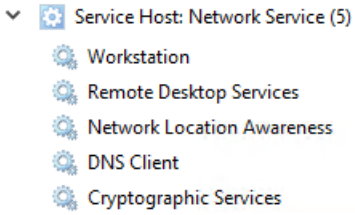
Try This

To see the refactoring behavior, create a Windows 10 version 1703 VM and configure the memory settings as follows:

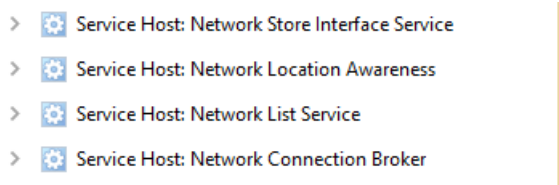
1. To see grouped processes, set the RAM to 3484 MB or less. Restart the VM and then open Task Manager.
2. To see separated processes, set the RAM to 3486 MB or greater. Restart the VM and then open Task Manager.

Refactoring also makes it easier to view running processes in Task Manager. You can look at Task Manager and know exactly which service is using what resources, without having to expand many separate host groups.

For example, here are the running processes displayed in Task Manager in Windows 10 version 1607:



Compare that to the same view of running processes in Windows 10 version 1703:



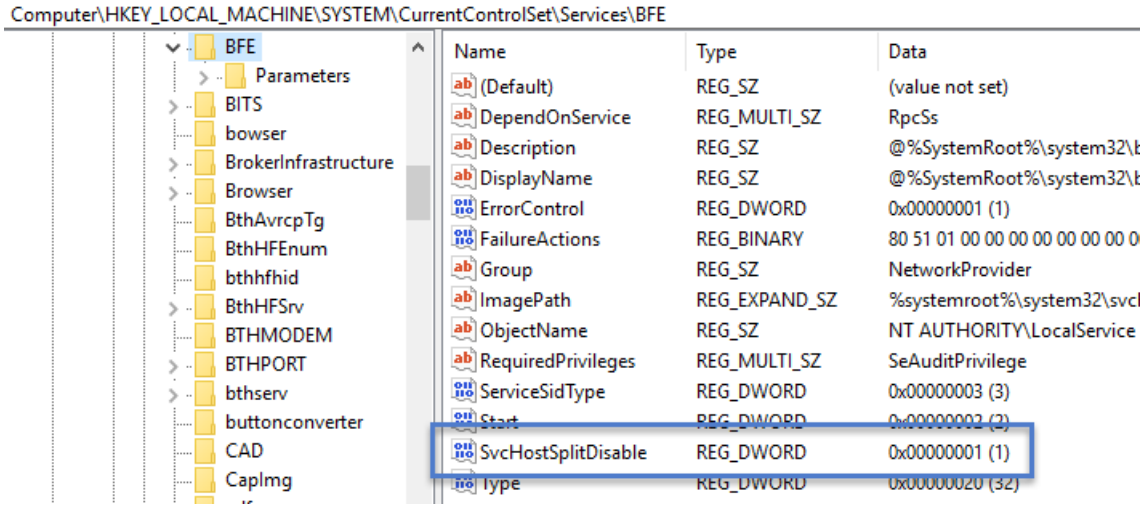
Exceptions

Some services will continue to be grouped on PCs running with 3.5GB or higher RAM. For example, the Base Filtering Engine (BFE) and the Windows Firewall (Mpssvc) will be grouped together in a single host group, as will the RPC Endpoint Mapper and Remote Procedure Call services.

If you need to identify services that will continue to be grouped, in addition to seeing them in Task Manager and using command line tools, you can look for the *SvcHostSplitDisable* value in their respective service keys under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services.

The default value of **1** prevents the service from being split.

For example, this is the registry key configuration for BFE:



Memory footprint

Be aware that separating services increases the total number of SvcHost instances, which increases memory utilization. (Service grouping provided a modest reduction to the overall resource footprint of the services involved.)

Consider the following:

GROUPED SERVICES (< 3.5GB)	SPLIT SERVICES (3.5GB+)
Name Count Mem (KB) ---- ----- -------- svchost 21 465,880	Name Count Mem (KB) ---- ----- -------- svchost 74 949,980

> **NOTE**
>
> The above represents the peak observed values.

The total number of service instances and the resulting memory utilization varies depending on activity. Instance counts can typically range from approximately 17-21 for grouped services, and 67-74 for separated services.

Try This

To determine the impact of splitting hosted services on a Windows 10 version 1703 PC, run the following Windows PowerShell cmdlet, before and after toggling the memory settings:

```
Get-Process SvcHost | Group-Object -Property ProcessName | Format-Table Name, Count, @{n='Mem (KB)';e=
{'{0:N0}' -f (($_.Group|Measure-Object WorkingSet -Sum).Sum / 1KB)};a='right'} -AutoSize
```

Per-user services in Windows 10 and Windows Server

6/26/2019 • 7 minutes to read • Edit Online

Applies to: Windows 10, Windows Server

Per-user services are services that are created when a user signs into Windows or Windows Server and are stopped and deleted when that user signs out. These services run in the security context of the user account - this provides better resource management than the previous approach of running these kinds of services in Explorer, associated with a preconfigured account, or as tasks.

> **NOTE**
> Per-user services are only in available in Windows Server if you have installed the Desktop Experience. If you are running a Server Core or Nano Server installation, you won't see these services.

You can set the template service's **Startup Type** to **Disabled** to create per-user services in a stopped and disabled state.

> **IMPORTANT**
> Carefully test any changes to the template service's Startup Type before deploying to a production environment.

Use the following information to understand per-user services, change the template service Startup Type, and manage per-user services through Group Policy and security templates. For more information about disabling system services for Windows Server, see Guidance on disabling system services on Windows Server with Desktop Experience.

Per-user services

The following table lists per-user services and when they were added to Windows 10 and Windows Server with the Desktop Experience. The template services are located in the registry at HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services.

Before you disable any of these services, review the **Description** column in this table to understand the implications, including dependent apps that will no longer work correctly.

WINDOWS VERSION	KEY NAME	DISPLAY NAME	DEFAULT START TYPE	DEPENDENCIES	DESCRIPTION
1803	BcastDVRUserService	GameDVR and Broadcast User Service	Manual		Used for Game Recordings and Live Broadcasts
1803	BluetoothUserService	Bluetooth User Support Service	Manual		Supports proper functionality of Bluetooth features relevant to each user session

WINDOWS VERSION	KEY NAME	DISPLAY NAME	DEFAULT START TYPE	DEPENDENCIES	DESCRIPTION
1803	CaptureService	CaptureService	Manual		OneCore Capture Service
1607	CDPUserSvc	CDPUserSvc	Auto	- Network Connection Broker - Remote Procedure Call (RPC) - TCP/IP Protocol Driver	Used for Connected Devices Platform scenarios
1803	DevicePickerUserSvc	DevicePicker	Manual		Device Picker
1703	DevicesFlowUserSvc	DevicesFlow	Manual		Device Discovery and Connecting
1703	MessagingService	MessagingService	Manual		Service supporting text messaging and related functionality
1607	OneSyncSvc	Sync Host	Auto (delayed)		Synchronizes mail, contacts, calendar, and other user data. Mail and other applications dependent on this service don't work correctly when this service is not running.
1607	PimIndexMaintenanceSvc	Contact Data	Manual	UnistoreSvc	Indexes contact data for fast contact searching. If you stop or disable this service, search results might not display all contacts.
1709	PrintWorkflowUserSvc	PrintWorkflow	Manual		Print Workflow

WINDOWS VERSION	KEY NAME	DISPLAY NAME	DEFAULT START TYPE	DEPENDENCIES	DESCRIPTION
1607	UnistoreSvc	User Data Storage	Manual		Handles storage of structured user data, including contact info, calendars, and messages. If you stop or disable this service, apps that use this data might not work correctly.
1607	UserDataSvc	User Data Access	Manual	UnistoreSvc	Provides apps access to structured user data, including contact info, calendars, and messages. If you stop or disable this service, apps that use this data might not work correctly.
1607	WpnUserService	Windows Push Notifications User Service	Manual		Hosts Windows notification platform, which provides support for local and push notifications. Supported notifications are tile, toast, and raw.

Disable per-user services

The template service isn't displayed in the Services console (services.msc) so you need to edit the registry directly, either with Group Policy or a scripted solution, to disable a per-user service.

> **NOTE**
>
> Disabling a per-user service simply means that it is created in a stopped and disabled state. When the user signs out, the per-user service is removed.

You can't manage all of the per-user service templates services using normal Group Policy management methods. Because the per-user services aren't displayed in the Services management console, they're also not displayed in the Group Policy Services policy editor UI.

Additionally, there are four template services that can't be managed with a security template:

- PimIndexMaintenanceSvc
- UnistoreSvc
- UserDataSvc

- WpnUserService

In light of these restrictions, you can use the following methods to manage per-user services template services:

- A combination of a security template and a script or Group Policy preferences registry policy
- Group Policy preferences for all of the services
- A script for all of the services

Manage template services using a security template

You can manage the CDPUserSvc and OneSyncSvc per-user services with a security template. See Administer security policy settings for more information.

For example:

```
[Unicode]
Unicode=yes
[Version]
signature="$CHICAGO$"
Revision=1
[Service General Setting]
"CDPUserSVC".4,""
```

Manage template services using Group Policy preferences

If a per-user service can't be disabled using a the security template, you can disable it by using Group Policy preferences.

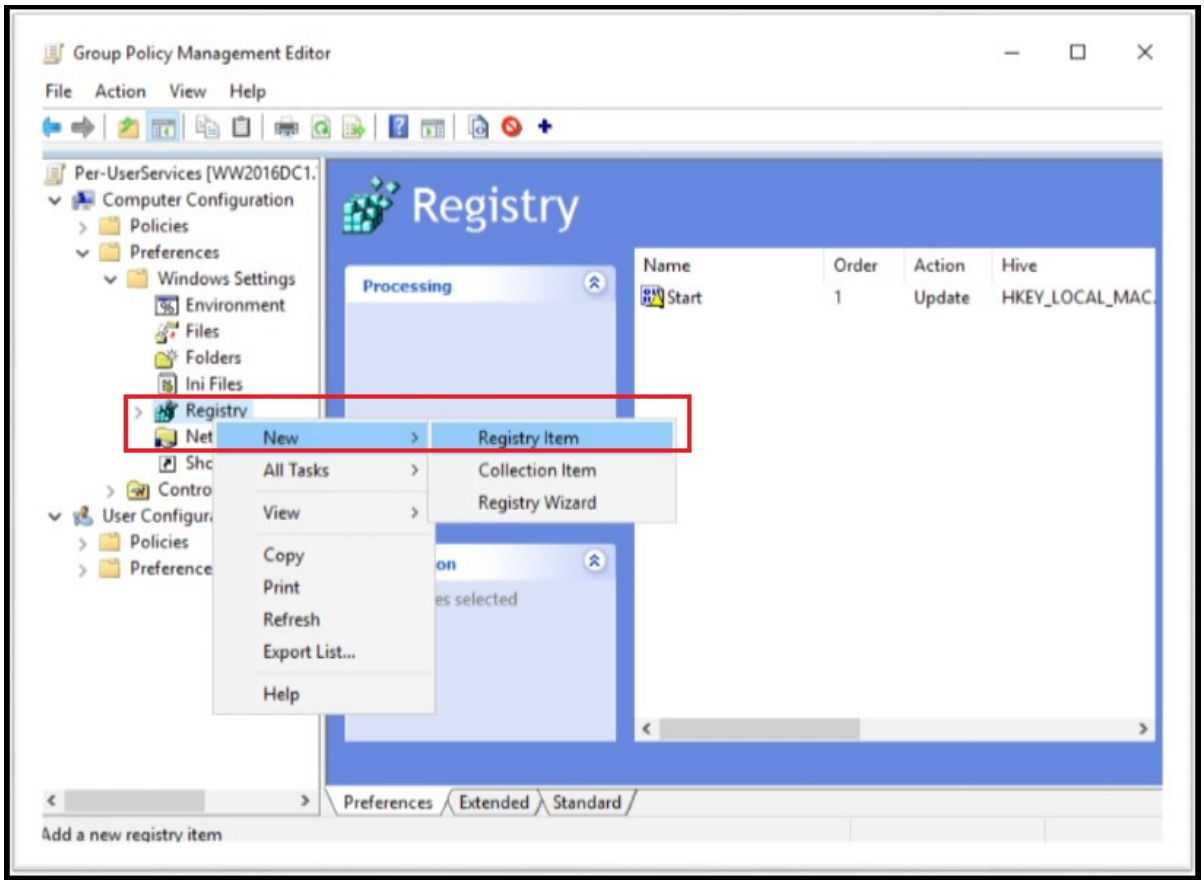
1. On a Windows Server domain controller or Windows 10 PC that has the Remote Server Administration Tools (RSAT) installed, click **Start**, type GPMC.MSC, and then press **Enter** to open the **Group Policy Management Console**.

2. Create a new Group Policy Object (GPO) or use an existing GPO.

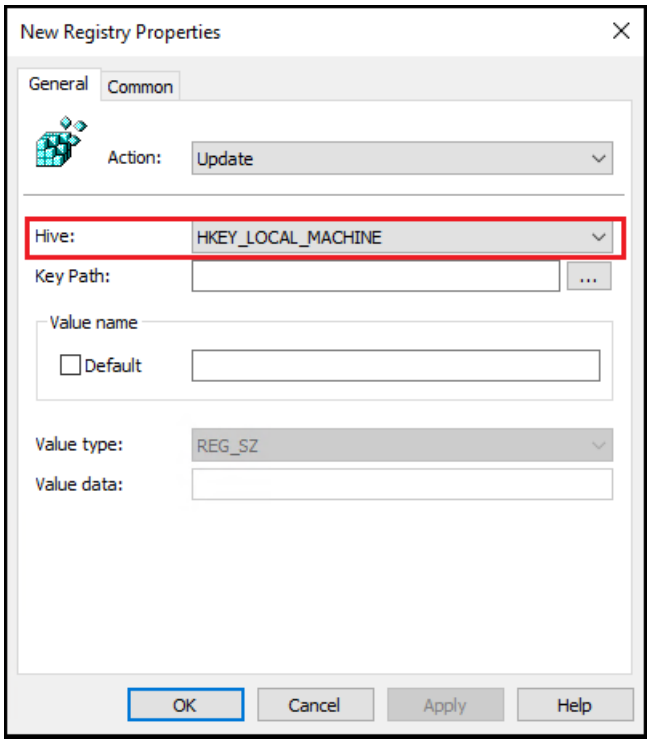
3. Right-click the GPO and click **Edit** to launch the Group Policy Object Editor.

4. Depending on how you want to target the Group Policy, under **Computer configuration** or **User configuration** browse to Preferences\Windows Settings\Registry.

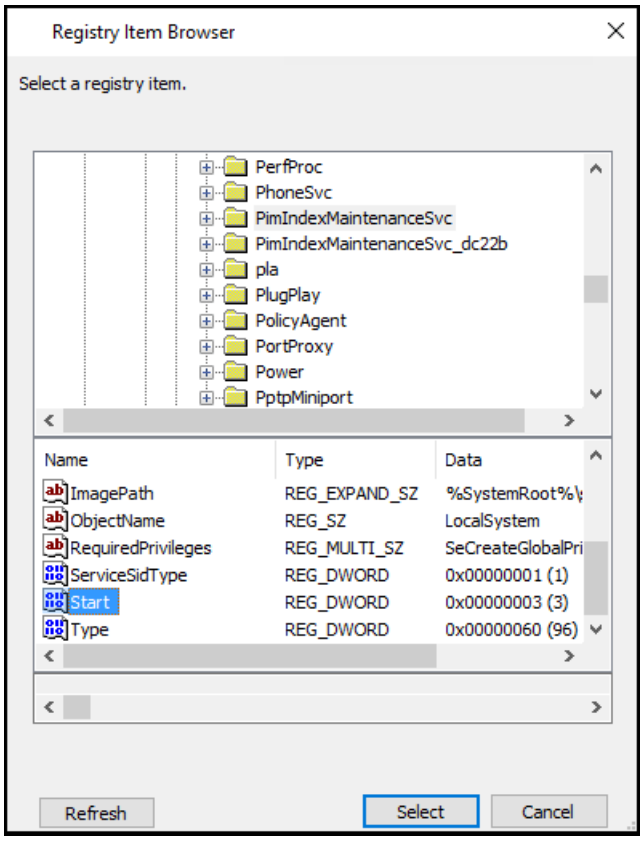
5. Right-click **Registry** > **New** > **Registry Item**.



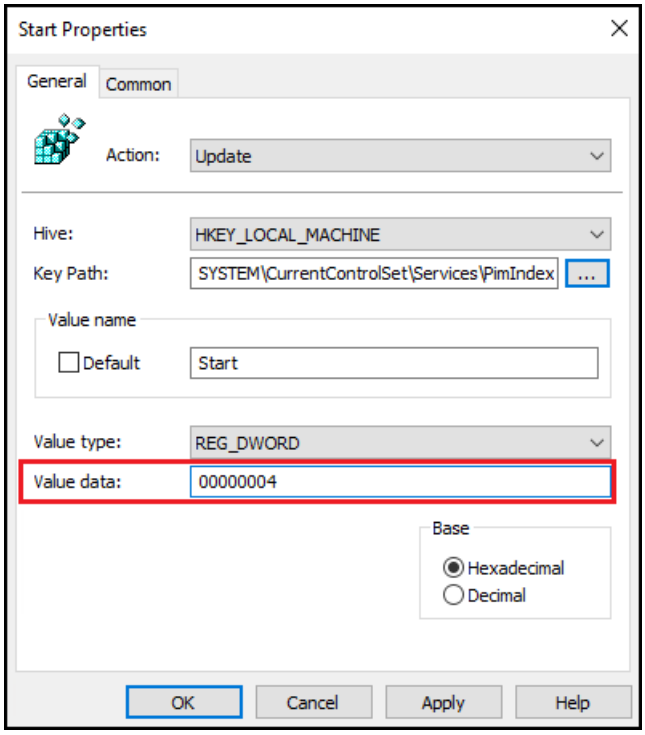
6. Make sure that HKEY_Local_Machine is selected for Hive and then click ... (the ellipses) next to Key Path.



7. Browse to **System\CurrentControlSet\Services\PimIndexMaintenanceSvc**. In the list of values, highlight **Start** and click **Select**.



8. Change **Value data** from **00000003** to **00000004** and click **OK**. Note setting the Value data to **4** = **Disabled**.



9. To add the other services that cannot be managed with a Group Policy templates, edit the policy and repeat steps 5-8.

Managing Template Services with reg.exe

If you cannot use Group Policy Preferences to manage the per-user services, you can edit the registry with reg.exe. To disable the Template Services, change the Startup Type for each service to 4 (disabled). For example:

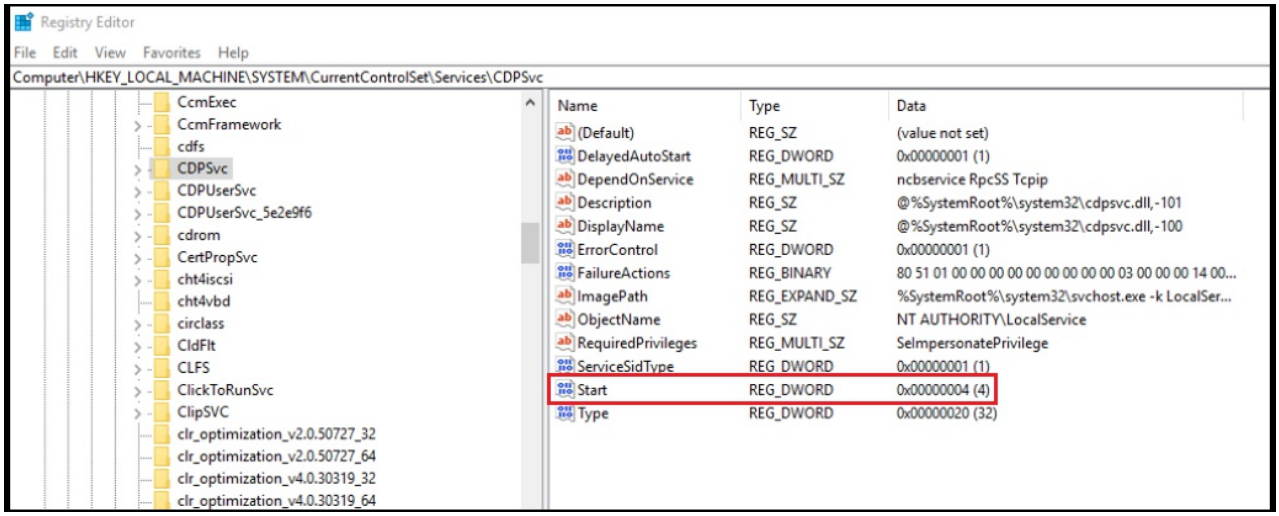
```
REG.EXE ADD HKLM\System\CurrentControlSet\Services\CDPUserSvc /v Start /t REG_DWORD /d 4 /f
REG.EXE ADD HKLM\System\CurrentControlSet\Services\OneSyncSvc /v Start /t REG_DWORD /d 4 /f
REG.EXE ADD HKLM\System\CurrentControlSet\Services\PimIndexMaintenanceSvc /v Start /t REG_DWORD /d 4 /f
REG.EXE ADD HKLM\System\CurrentControlSet\Services\UnistoreSvc /v Start /t REG_DWORD /d 4 /f
REG.EXE ADD HKLM\System\CurrentControlSet\Services\UserDataSvc /v Start /t REG_DWORD /d 4 /f
REG.EXE ADD HKLM\System\CurrentControlSet\Services\WpnUserService /v Start /t REG_DWORD /d 4 /f
```

Caution

We recommend that you do not directly edit the registry unless there is no other alternative. Modifications to the registry are not validated by the Registry Editor or by the Windows operating system before they are applied. As a result, incorrect values can be stored, and this can result in unrecoverable errors in the system. When possible, instead of editing the registry directly, use Group Policy or other Windows tools such as the Microsoft Management Console (MMC) to accomplish tasks. If you must edit the registry, use extreme caution.

Managing Template Services with regedit.exe

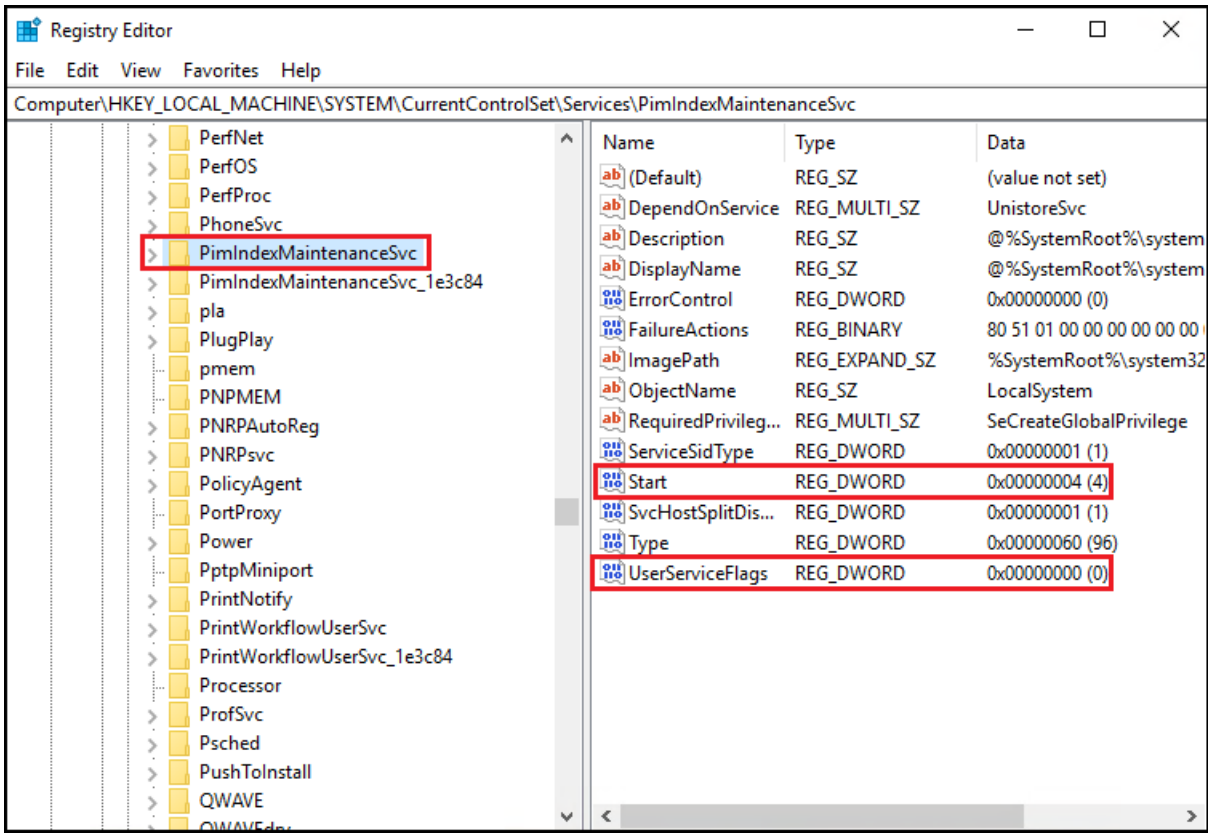
If you cannot use Group Policy preferences to manage the per-user services, you can edit the registry with regedit.exe. To disable the template services, change the Startup Type for each service to 4 (disabled):



Caution

We recommend that you do not directly edit the registry unless there is no other alternative. Modifications to the registry are not validated by the Registry Editor or by the Windows operating system before they are applied. As a result, incorrect values can be stored, and this can result in unrecoverable errors in the system. When possible, instead of editing the registry directly, use Group Policy or other Windows tools such as the Microsoft Management Console (MMC) to accomplish tasks. If you must edit the registry, use extreme caution.

Beginning with Windows 10, version 1709 and Windows Server, version 1709, you can prevent the per-user service from being created by setting **UserServiceFlags** to 0 under the same service configuration in the registry:



Manage template services by modifying the Windows image

If you're using custom images to deploy Windows, you can modify the Startup Type for the template services as part of the normal imaging process.

Use a script to manage per-user services

You can create a script to change the Startup Type for the per-user services. Then use Group Policy or another management solution to deploy the script in your environment.

Sample script using sc.exe:

```
sc.exe configure <service name> start= disabled
```

Note that the space after "=" is intentional.

Sample script using the Set-Service PowerShell cmdlet:

```
Set-Service <service name> -StartupType Disabled
```

View per-user services in the Services console (services.msc)

As mentioned you can't view the template services in the Services console, but you can see the user-specific per-user services - they are displayed using the <service name>_LUID format (where LUID is the locally unique identifier).

For example, you might see the following per-user services listed in the Services console:

- CPDUserSVC_443f50
- ContactData_443f50
- Sync Host_443f50
- User Data Access_443f50
- User Data Storage_443f50

View per-user services from the command line

You can query the service configuration from the command line. The **Type** value indicates whether the service is a user-service template or user-service instance.

```
PS C:\> sc.exe qc BcastDVRUserService
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: BcastDVRUserService
        TYPE               : 60  USER_SHARE_PROCESS TEMPLATE
        START_TYPE         : 3   DEMAND_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : C:\Windows\system32\svchost.exe -k BcastDVRUserService
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : GameDVR and Broadcast User Service
        DEPENDENCIES       : RpcSs
        SERVICE_START_NAME : LocalSystem
PS C:\> sc.exe qc BcastDVRUserService_660dc
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: BcastDVRUserService_660dc
        TYPE               : e0  USER_SHARE_PROCESS INSTANCE
        START_TYPE         : 3   DEMAND_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : C:\Windows\system32\svchost.exe -k BcastDVRUserService
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : GameDVR and Broadcast User Service_660dc
        DEPENDENCIES       :
        SERVICE_START_NAME :
```

Deploy application upgrades on Windows 10 Mobile

6/10/2019 • 2 minutes to read • Edit Online

Applies to: Windows 10

When you have a new version of an application, how do you get that to the Windows 10 Mobile devices in your environment? With application supersedence in System Center Configuration Manager.

There are two steps to deploy an app upgrade:

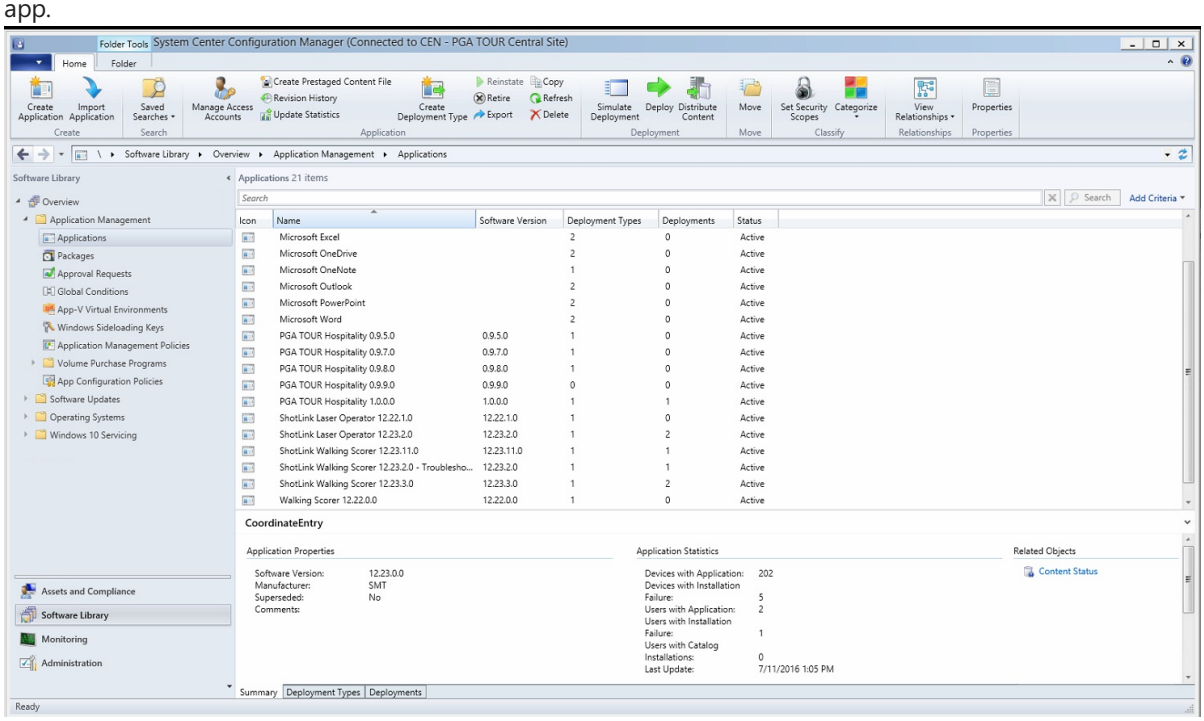
1. Define the supersedence - this lets Configuration Manager know that the old version should be replaced by the new version.
2. Deploy the upgrade to your users.

The following steps walk you through the upgrade deployment process - we have an upgraded version of the Walking Scorer app (moving from version 12.23.2.0 to 12.23.3.0). Because we previously used Configuration Manager to deploy the existing version, we'll use it now to upgrade the app.

Before you can deploy the upgrade, make sure you import the new version of the app and distribute it to your manage.microsoft.com distribution point.

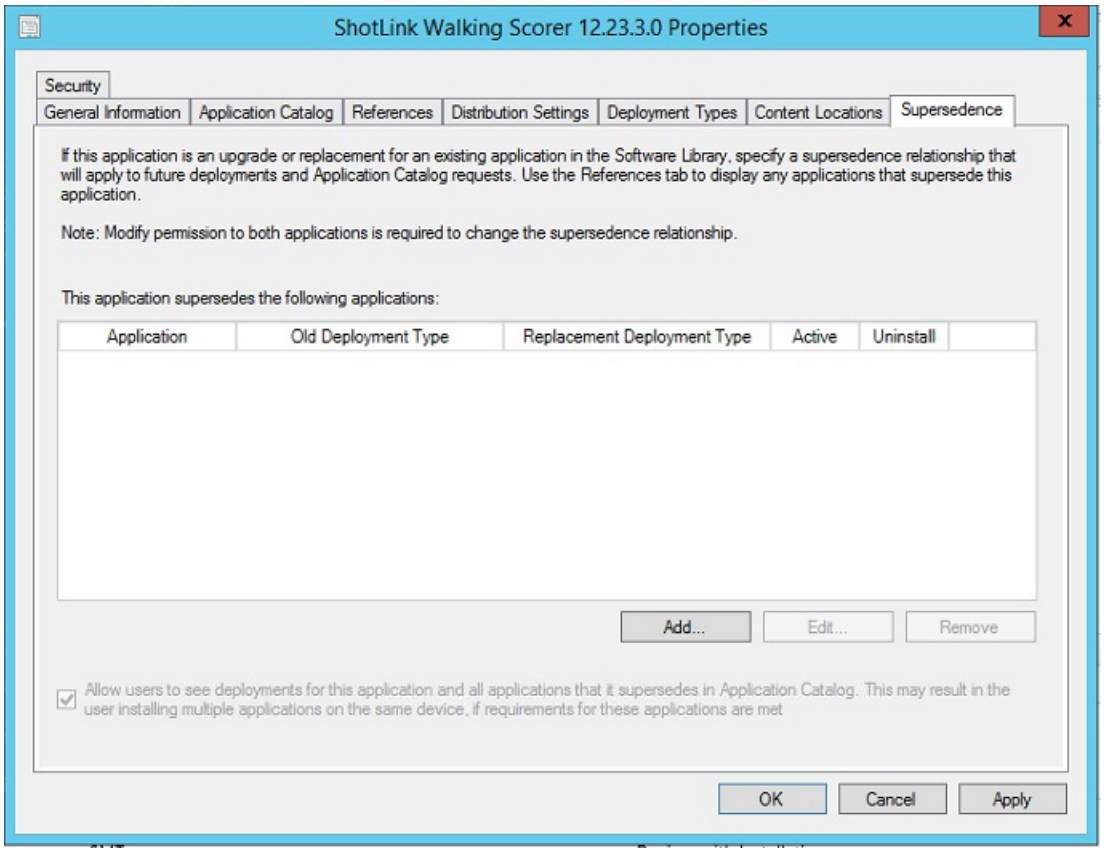
Define app supersedence

1. In the Configuration Manager console, open the Software Library, and then find the new version of your app.



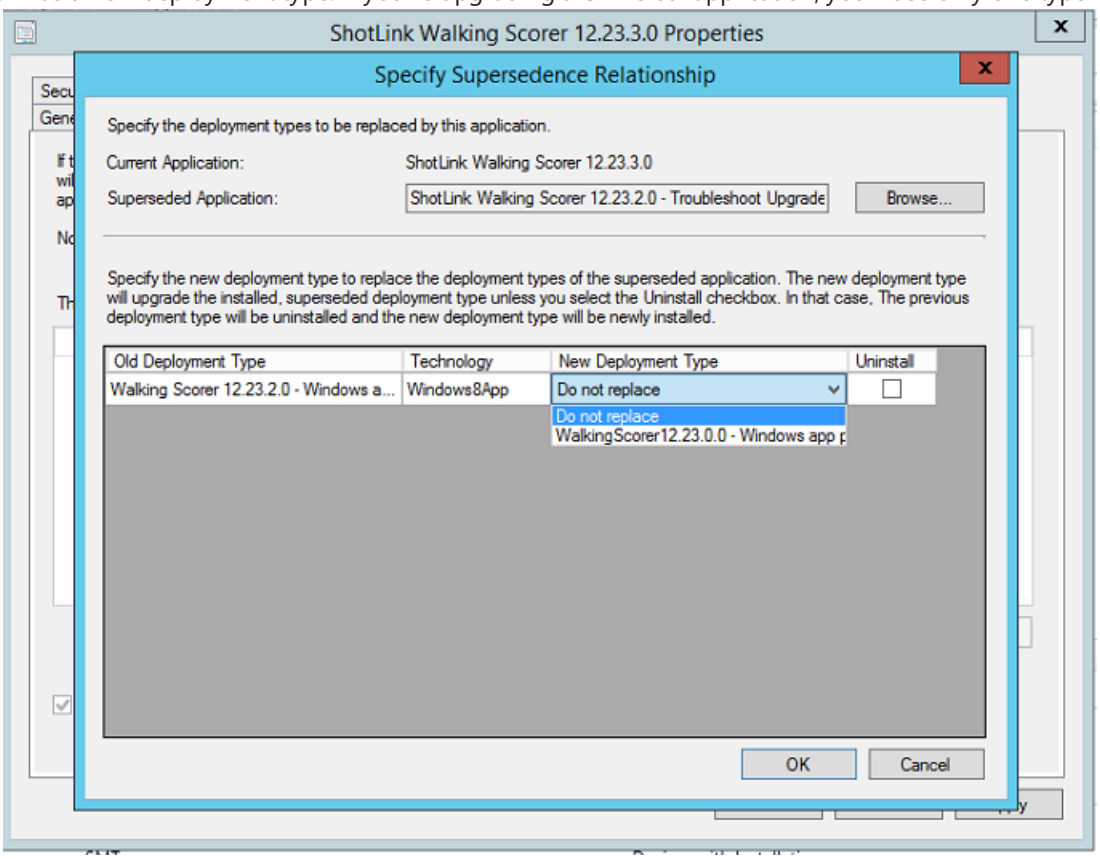
2. Right-click the new version, and then click **Properties**.

3. Click the **Supersedence** tab - there shouldn't be any supersedence rules yet. We'll add one next.



4. Click **Add**, browse to the existing (older) version of the app that you're upgrading, and then click **OK**.

5. Under **New Deployment Type** select the new version of the app. (When you imported the new version, it comes in as a new deployment type. If you're upgrading a Universal application, you'll see only one type here.)



> **IMPORTANT**
>
> Do **NOT** select **Uninstall**. This tells Configuration Manager to uninstall the old version, but it does **NOT** then install the new version.

6. Click **OK**.

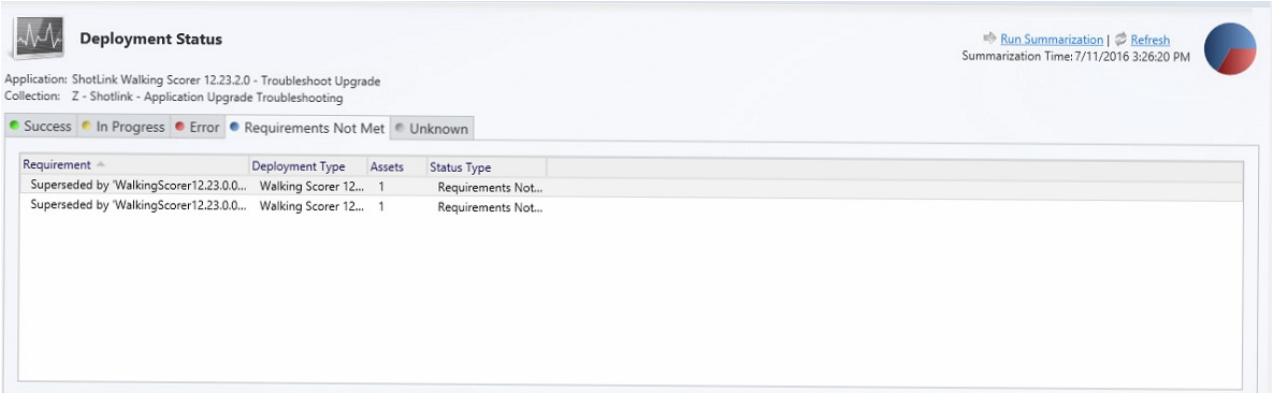
7. If you have other versions of the same app, repeat steps 4-6 for each version. Click **OK** when you're done.

> **NOTE**
>
> Need to remove a supersedence? (Maybe the new version turned out to be flaky and you don't want users to get it yet.) On the **Supersedence** tab for the *new* version of the app, double-click the older version in the list of supersedence rules, and then change the **New Deployment Type** to **Do not replace**.

Deploy the app upgrade

You're now ready to deploy the upgrade. On the **Home** tab in Configuration Manager, select the new version of the app, and then click **Deploy**, and follow the instructions in the wizard. When asked, set the **Purpose** to **Required**.

You don't need to delete the deployment associated with the older version of the app. The status for that deployment will change to **Requirements not met** in the **Monitoring** view:



If you haven't deployed an app through Configuration Manager before, check out Deploy applications with System Center Configuration Manager. You can also see how to delete deployments (although you don't have to) and notify users about the upgraded app.

Change history for Application management in Windows 10

5/31/2019 • 2 minutes to read • Edit Online

This topic lists new and updated topics in the Configure Windows 10 documentation for Windows 10 and Windows 10 Mobile.

RELEASE: Windows 10, version 1803

The topics in this library have been updated for Windows 10, version 1803.

October 2017

NEW OR CHANGED TOPIC	DESCRIPTION
Enable or block Windows Mixed Reality apps in the enterprise	Added instructions for manually installing Windows Mixed Reality

RELEASE: Windows 10, version 1709

The topics in this library have been updated for Windows 10, version 1709 (also known as the Fall Creators Update). The following new topic has been added:

- Enable or block Windows Mixed Reality apps in the enterprise

September 2017

NEW OR CHANGED TOPIC	DESCRIPTION
Per-user services in Windows 10	New
Remove background task resource restrictions	New
Understand the different apps included in Windows 10	New

July 2017

NEW OR CHANGED TOPIC	DESCRIPTION
Service Host process refactoring	New
Deploy app upgrades on Windows 10 Mobile	New

How to keep apps removed from Windows 10 from returning during an update

6/6/2019 • 3 minutes to read • Edit Online

Applies to: Windows 10 (Semi-Annual Channel)

When you update a computer running Windows 10, version 1703 or 1709, you might see provisioned apps that you previously removed return post-update. This can happen if the computer was offline when you removed the apps. This issue was fixed in Windows 10, version 1803.

> **NOTE**
>
> - This issue only occurs after a feature update (from one version to the next), not monthly updates or security-related updates.
> - This only applies to first-party apps that shipped with Windows 10. This doesn't apply to third-party apps, Microsoft Store apps, or LOB apps.
> - This issue can occur whether you removed the app using `Remove-appxprovisionedpackage` or `Get-AppxPackage -allusers | Remove-AppxPackage -Allusers`.

To remove a provisioned app, you need to remove the provisioning package. The apps might reappear if you removed the packages in one of the following ways:

- If you removed the packages while the wim file was mounted when the device was offline.
- If you removed the packages by running a PowerShell cmdlet on the device while Windows was online. Although the apps won't appear for new users, you'll still see the apps for the user account you signed in as.

When you remove a provisioned app, we create a registry key that tells Windows not to reinstall or update that app the next time Windows is updated. If the computer isn't online when you deprovision the app, then we don't create that registry key. (This behavior is fixed in Windows 10, version 1803. If you're running Windows 10, version 1709, apply the latest security update to fix it.)

> **NOTE**
>
> If you remove a provisioned app while Windows is online, it's only removed for *new users*—the user that you signed in as will still have that provisioned app. That's because the registry key created when you deprovision the app only applies to new users created *after* the key is created. This doesn't happen if you remove the provisioned app while Windows is offline.

To prevent these apps from reappearing at the next update, manually create a registry key for each app, then update the computer.

Create registry keys for deprovisioned apps

Use the following steps to create a registry key:

1. Identify any provisioned apps you want removed. Record the package name for each app.
2. Create a .reg file to generate a registry key for each app. Use this list of Windows 10, version 1709 registry keys as your starting point.
   a. Paste the list of registry keys into Notepad (or a text editor).

b. Remove the registry keys belonging to the apps you want to keep. For example, if you want to keep the Bing Weather app, delete this registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\A
ppxAllUserStore\Deprovisioned\Microsoft.BingWeather_8wekyb3d8bbwe]
```

c. Save the file with a .txt extension, then right-click the file and change the extension to .reg.

3. Double-click the .reg file to create the registry keys. You can see the new keys in HKLM\path-to-reg-keys.

You're now ready to update your computer. After the update, check the list of apps in the computer to confirm the removed apps are still gone.

Package names for apps provisioned in Windows 10, version 1709

DISPLAYED APP NAME	PACKAGE NAME
Microsoft.3DBuilder	Microsoft.3DBuilder_15.2.10821.1000_neutral_~_8wekyb3d8bbwe
Microsoft.BingWeather	Microsoft.BingWeather_4.23.10923.0_neutral_~_8wekyb3d8bbwe
Microsoft.DesktopAppInstaller	Microsoft.DesktopAppInstaller_1.10.16004.0_neutral_~_8wekyb3d8bbwe
Microsoft.GetHelp	Microsoft.GetHelp_10.1706.1811.0_neutral_~_8wekyb3d8bbwe
Microsoft.Getstarted	Microsoft.Getstarted_5.12.2691.1000_neutral_~_8wekyb3d8bbwe
Microsoft.HEVCVideoExtension	Microsoft.HEVCVideoExtension_1.0.2512.0_x64__8wekyb3d8bbwe
Microsoft.Messaging	Microsoft.Messaging_2018.124.707.0_neutral_~_8wekyb3d8bbwe
Microsoft.Microsoft3DViewer	Microsoft.Microsoft3DViewer_3.1803.29012.0_neutral_~_8wekyb3d8bbwe
Microsoft.MicrosoftOfficeHub	Microsoft.MicrosoftOfficeHub_2017.715.118.0_neutral_~_8wekyb3d8bbwe
Microsoft.MicrosoftSolitaireCollection	Microsoft.MicrosoftSolitaireCollection_3.18.12091.0_neutral_~_8wekyb3d8bbwe
Microsoft.MicrosoftStickyNotes	Microsoft.MicrosoftStickyNotes_2.1.18.0_neutral_~_8wekyb3d8bbwe
Microsoft.MSPaint	Microsoft.MSPaint_4.1803.21027.0_neutral_~_8wekyb3d8bbwe
Microsoft.Office.OneNote	Microsoft.Office.OneNote_2015.9126.21251.0_neutral_~_8wekyb3d8bbwe

DISPLAYED APP NAME	PACKAGE NAME
Microsoft.OneConnect	Microsoft.OneConnect_3.1708.2224.0_neutral_~_8wekyb3d8bbwe
Microsoft.People	Microsoft.People_2017.1006.1846.1000_neutral_~_8wekyb3d8bbwe
Microsoft.Print3D	Microsoft.Print3D_1.0.2422.0_neutral_~_8wekyb3d8bbwe
Microsoft.SkypeApp	Microsoft.SkypeApp_12.1811.248.1000_neutral_~_kzf8qxf38zg5c
Microsoft.StorePurchaseApp	Microsoft.StorePurchaseApp_11802.1802.23014.0_neutral_~_8wekyb3d8bbwe
Microsoft.Wallet	Microsoft.Wallet_1.0.16328.0_neutral_~_8wekyb3d8bbwe
Microsoft.Windows.Photos	Microsoft.Windows.Photos_2018.18022.15810.1000_neutral_~_8wekyb3d8bbwe
Microsoft.WindowsAlarms	Microsoft.WindowsAlarms_2017.920.157.1000_neutral_~_8wekyb3d8bbwe
Microsoft.WindowsCalculator	Microsoft.WindowsCalculator_2017.928.0.1000_neutral_~_8wekyb3d8bbwe
Microsoft.WindowsCamera	Microsoft.WindowsCamera_2017.1117.10.1000_neutral_~_8wekyb3d8bbwe
microsoft.windowscommunicationsapps	microsoft.windowscommunicationsapps_2015.9126.21425.0_neutral_~_8wekyb3d8bbwe
Microsoft.WindowsFeedbackHub	Microsoft.WindowsFeedbackHub_2018.323.50.1000_neutral_~_8wekyb3d8bbwe
Microsoft.WindowsMaps	Microsoft.WindowsMaps_2017.1003.1829.1000_neutral_~_8wekyb3d8bbwe
Microsoft.WindowsSoundRecorder	Microsoft.WindowsSoundRecorder_2017.928.5.1000_neutral_~_8wekyb3d8bbwe
Microsoft.WindowsStore	Microsoft.WindowsStore_11803.1001.613.0_neutral_~_8wekyb3d8bbwe
Microsoft.Xbox.TCUI	Microsoft.Xbox.TCUI_1.8.24001.0_neutral_~_8wekyb3d8bbwe
Microsoft.XboxApp	Microsoft.XboxApp_39.39.21002.0_neutral_~_8wekyb3d8bbwe
Microsoft.XboxGameOverlay	Microsoft.XboxGameOverlay_1.24.5001.0_neutral_~_8wekyb3d8bbwe
Microsoft.XboxIdentityProvider	Microsoft.XboxIdentityProvider_2017.605.1240.0_neutral_~_8wekyb3d8bbwe

DISPLAYED APP NAME	PACKAGE NAME
Microsoft.XboxSpeechToTextOverlay	Microsoft.XboxSpeechToTextOverlay_1.21.13002.0_neutral_~_8wekyb3d8bbwe
Microsoft.ZuneMusic	Microsoft.ZuneMusic_2019.18011.13411.1000_neutral_~_8wekyb3d8bbwe
Microsoft.ZuneVideo	Microsoft.ZuneVideo_2019.17122.16211.1000_neutral_~_8wekyb3d8bbwe

Registry keys for provisioned apps

```
Windows Registry Editor Version 5.00
;1709 Registry Keys

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.BingWeather_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.DesktopAppInstaller_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.GetHelp_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.Getstarted_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.Microsoft3DViewer_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.MicrosoftOfficeHub_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.MicrosoftSolitaireCollection_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.MicrosoftStickyNotes_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.MSPaint_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.Office.OneNote_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.OneConnect_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.People_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.Print3D_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.SkypeApp_kzf8qxf38zg5c]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.StorePurchaseApp_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.Wal
```

```
let_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.Win
dows.Photos_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.Win
dowsAlarms_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.Win
dowsCalculator_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.Win
dowsCamera_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\microsoft.win
dowscommunicationsapps_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.Win
dowsFeedbackHub_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.Win
dowsMaps_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.Win
dowsSoundRecorder_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.Win
dowsStore_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.Xbo
x.TCUI_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.Xbo
xApp_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.Xbo
xGameOverlay_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.Xbo
xIdentityProvider_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.Xbo
xSpeechToTextOverlay_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.Zun
eMusic_8wekyb3d8bbwe]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Appx\AppxAllUserStore\Deprovisioned\Microsoft.Zun
eVideo_8wekyb3d8bbwe]
```

Get-AppxPackage Get-AppxPackage -allusers Remove-AppxPackage