

**Microsoft Application Virtualization 5.0**

**Sequencing Guide**

White Paper Descriptor

This whitepaper is designed to provide administrators with guidance for sequencing applications to create virtual packages that can be delivered to the end user. This document discusses setting up the App-V Sequencer, sequencing best practices, examples of sequencing, important information related to updating packages, and examples of scripting.

Copyright © 2012 MICROSOFT CORPORATION

Contents

[1 Introduction to Application Sequencing 5](#_Toc338432783)

[2 Sequencer Workstation Configuration 6](#_Toc338432784)

[3 Recommended Best Practices for Sequencing 7](#_Toc338432785)

[3.1 Items to document in a recipe include: 7](#_Toc338432786)

[3.2 Recommended best practices: 7](#_Toc338432787)

[4 Classifying Applications for Sequencing 9](#_Toc338432788)

[5 Sequencing Limitations 10](#_Toc338432789)

[6 Sequence files 11](#_Toc338432790)

[7 Sequencing Walk Through 12](#_Toc338432791)

[7.1 Welcome Screen 12](#_Toc338432792)

[7.2 Packaging Method 13](#_Toc338432793)

[7.3 Prepare Computer 14](#_Toc338432794)

[7.4 Type of Application 15](#_Toc338432795)

[7.5 Select Installer 16](#_Toc338432796)

[7.6 Package Name 17](#_Toc338432797)

[7.7 Installation 18](#_Toc338432798)

[7.7.1 Applications that Require a Reboot 18](#_Toc338432799)

[7.8 Configure Software 19](#_Toc338432800)

[7.9 Installation Report 20](#_Toc338432801)

[7.10 Customize 21](#_Toc338432802)

[7.11 Prepare for Streaming 22](#_Toc338432803)

[7.12 Target OS 24](#_Toc338432804)

[7.13 Create Package 25](#_Toc338432805)

[7.14 Completion Report 26](#_Toc338432806)

[7.15 Package Editor 27](#_Toc338432807)

[7.16 Sequencing with PowerShell 35](#_Toc338432808)

[8 Creating a Package Accelerator 36](#_Toc338432809)

[8.1 Work-flow 37](#_Toc338432810)

[8.2 Creating a Package Accelerator with PowerShell 41](#_Toc338432811)

[9 Create a package using a Package Accelerator 42](#_Toc338432812)

[9.1 Work-flow 42](#_Toc338432813)

[9.2 Create a package using a Package Accelerator with PowerShell 43](#_Toc338432814)

[10 Application Package Upgrade 44](#_Toc338432815)

[10.1 Updating a Package that will Replace the Existing Package 45](#_Toc338432816)

[10.2 Updating a Package for Deployment with the Existing Package 46](#_Toc338432817)

[10.3 Update an existing package with PowerShell 47](#_Toc338432818)

[11 Sequencing for Connection Groups (Plug-ins and Middleware) 48](#_Toc338432819)

[11.1 Advantages 48](#_Toc338432820)

[11.2 Workflow 49](#_Toc338432821)

[11.3 Add-on or Plug-in 50](#_Toc338432822)

[11.4 Middleware 51](#_Toc338432823)

[12 Sequencer Templates 52](#_Toc338432824)

[13 App-V Package Converter 53](#_Toc338432825)

[13.1 Package Converter PowerShell commands 53](#_Toc338432826)

[14 Dynamic Configuration and Targeted Scripting 54](#_Toc338432827)

[14.1 Deployment Configuration Files 54](#_Toc338432828)

[14.1.1 DeploymentConfig vs. UserConfig. 54](#_Toc338432829)

[14.1.2 Comparing Script processing in Deployment and User Config files 55](#_Toc338432830)

[14.2 Enabling Scripting 56](#_Toc338432831)

[14.3 Writing a Script 57](#_Toc338432832)

[14.3.1 Referencing script collateral 57](#_Toc338432833)

[14.3.2 Blocking Logic 58](#_Toc338432834)

[14.4 Usage Scenario 59](#_Toc338432835)

[14.5 Script Examples. 59](#_Toc338432836)

[14.6 Complete settings available in the Dynamic Configuration files 61](#_Toc338432837)

[14.6.1 Sample *XmlNotepad\_*DeploymentConfig.xml 61](#_Toc338432838)

[15 Advanced Sequencing Techniques 66](#_Toc338432839)

[15.1 Sequencing Web-Based Applications 66](#_Toc338432840)

[16 Additional Information 67](#_Toc338432841)

[16.1 Built-in reporting, Error Codes, and the Microsoft Knowledge Base 67](#_Toc338432842)

[16.2 App-V Sequencer Tokens 68](#_Toc338432843)

[16.3 App-V Team Blog 71](#_Toc338432844)

[16.4 App-V Technical Discussion Forums and Web Sites 71](#_Toc338432845)

# Introduction to Application Sequencing

The Microsoft Application Virtualization (App-V) Sequencer is a component of the App-V suite used to package applications to be deployed to systems using the App-V Client. Properly sequencing applications is critical to a successful App-V implementation. As such, it is important to follow recommended practices from Microsoft and be aware of the different options when sequencing. This document covers setting up the Sequencer, sequencing best practices, an example of sequencing, updating, the new package conversion feature, and finally, examples of advanced configuration file scripting.

# Sequencer Workstation Configuration

**Proper configuration of the sequencing station is imperative to ensure that applications will function properly when launched on a client. Microsoft recommends the following configuration when sequencing:**

* **Ensure the Sequencer operating system meets the minimum requirements:**
  + Processor—1 GHz or faster. The sequencing process is a single- threaded process, and it does not take advantage of multiple cores or processors.
  + Memory—1 GB RAM or greater
  + 500-MB page file is recommended.
  + Hard drive 30-GB minimum. It is recommended that the disk drive be at least three times as large as the largest application that will be sequenced.
  + Windows® 7 SP1 or greater
* .NET 4.0 framework
* PowerShell 3.0

**Note:** Installation of any additional pre-requisites on the machine is not recommended unless it is for very advanced scenarios.

* **Sequence using a virtual machine.** Many applications will be sequenced more than once. This may be due to additional configuration changes or starting over to correct a mistake. Thus, the sequencing workstation needs to return to a clean configuration several times. Using a virtual machine is strongly advised to allow the use of snapshots and/or undo disks. After sequencing a virtual application package, the machine can be reverted to a clean state with a simple click of a button.
  + Install the latest Windows updates on the machine. Restart if required.
  + Install the Sequencer.
  + Apply a base snapshot (“base” state)
  + Return to the base snapshot before starting each new sequence.
* **Sequence on a machine that matches the operating system (OS) and configuration of the target clients**. It is often possible to sequence on one OS and run the virtualized application on a different OS; however this scenario is both application- and OS-dependent and is not guaranteed to work for all application/OS combinations since App-V is not a general-purpose OS compatibility solution. If problems are encountered, the application may require sequencing on the same OS environment that the App-V Client is running on in order to resolve those problems.

# Recommended Best Practices for Sequencing

Microsoft has best practices for sequencing to increase the number of successfully sequenced applications and reduce the sequencing time for packaging engineers.

The primary recommendation for all sequencing processes is for packaging engineers to familiarize themselves with the installation and execution of the application prior to sequencing. Make sure to read and become familiar with the installation instructions associated with the application so you understand how the application runs, as well as the components of the application the user will need.

Having step-by-step documentation on hand can help reduce unnecessary troubleshooting during the sequencing process by ensuring no important steps are skipped. Always document the sequencing process using a standard template, thereby creating a “recipe” that another packaging engineer in the organization can utilize to recreate the same package.

## Items to document in a recipe include:

* What application components are needed and will be required to complete the installation of the application?
* What updates, such as adding new files to the package (patches, templates, etc.), are required for successful installation of the application?
* What post-installation configuration steps are required for preparing the application for users?
* How do users commonly use this application immediately after its launch?
* Does this application do something that App-V currently [does not support](#_Sequencing_Limitations)? If so, check the Microsoft Knowledge Base to see if a workaround is available.

## Recommended best practices:

* **User Account Control (UAC):** Sequence with the same UAC setting that will be deployed to user desktops. For example; if UAC is disabled on client workstations, UAC should also be disabled on the sequencing workstation.
* **Extract self-extracting setup files to the %temp% directory of the sequencing machine for optimal package size.** Certain applications have EXE files that extract the installer to the machine before the initial install. To avoid adding unnecessary files to the sequence, it is a best practice to extract the installer onto the local machine prior to sequencing or to extract the installer to an excluded location.
* **Selecting the optimal Primary Virtual Application Directory (PVAD):** For best results, set the Primary Virtual Application Directory to match the directory the application is installed to for the following reasons:
  + Application Compatibility: some virtual applications will not function correctly, or even fail to launch, if the directories do not match.
  + Performance: no file system redirection is needed resulting in improved runtime performance.
  + Locate the application installation directory:
    - Open the application installer, and note the installation path. (Ex. Notepad is C:\Program Files\Notepad). Cancel the application installer. Launch the sequencer and insert the installation path into the Primary Virtual Application Directory.
    - Set the Primary Virtual Application Directory installation path (Ex. C:\Program Files\AppName). When the application installer loads, change its default installation path to match the Primary Virtual Application Directory (C:\Program Files\AppName).
* **Disable “Auto Update” features.** Some applications have the ability to check a web site or a server for the latest application updates. This feature should be turned off, as version control should be performed via sequencing new versions.
* **Disable “Install on First Use.”** Some applications have the option to “Install on First Use” for certain components. Do not sequence components with this option; choose either “Run from My Computer” (install this component) or “Not Available” (do not install this component). Application components that will not be used by any of the targeted users should not be installed.
* **Configure and test the application after installation.** Completing the installation of an application often requires performing several manual steps that are not part of the application installation process. These steps can involve configuring a connection to a ‘back-end’ database or copying updated files, etc. Do these configurations in the “Configure Software” phase, after checking the “I am finished installing” box and clicking “next” in the Sequencer. It is recommended to launch each application multiple times until the program is in a static state in the Installation Phase. For example, the sequencing engineer may need to run the application multiple times to get past all registration and dialog box requests. Some applications perform different tasks on first launch, second launch, and sometimes subsequent launches. Performing multiple launches will ensure completion of any required post-installation tasks (e.g., accepting a license agreement or setting file type associations).
* **Use the Description field** in the Packages tab in the Package Editor to note any details about the package that should be included. These notes will be valuable when revisiting the sequence later or when upgrading.
* **If deploying via streaming,** use the “Prepare for Streaming” section in the Wizard to launch each executable in a suite of applications. This will ensure that each application will have the required initial launch data on the App-V Client.
  + Operations made during the “Prepare for Streaming” section will be included in the primary feature block. When building the primary feature block, make sure to execute the application’s most common operations so they are included in the initial streaming of the application, creating an accurate primary feature block. If this is not done, users might see delays as they start to use the application and will regard it as being slow if many of the features they use are not in the primary feature block. Additionally, in an environment where bandwidth is limited, it is important to have an accurate primary feature block so users are not constantly making calls to the server to download additional files in cache.
* **Define target operating systems in the Deployment tab** only in situations where it is required, such as applications sequenced specifically for Remote Desktop Services (formerly Terminal Services) or when the sequenced package will only work on specific operating systems or platform types (32-bit vs 64-bit).

# Classifying Applications for Sequencing

All applications are different and therefore, no application will require the same amount of time to sequence. However, sequencing estimates can be put into three categories based on the complexity of the application, size (both size on disk and number of files), and reliance on resources outside of the virtual application. Another item to take special note of before sequencing begins: **nothing will slow down the sequencing process more than not having access to someone who inherently understands the full functionality of the application.**

**Note:** This section includes approximate sequencing times referring to first time sequencing and is not related to sequencing via the use of Package Accelerators. These estimates are very general. Every application will be different. The times are presented only to assist in *estimating* the time required for a first time non-accelerator project.

|  |  |  |
| --- | --- | --- |
| Application Type | Description | Time Scale |
| Simple | These applications are normally small. An example of an application in this category would be WinZip® or Adobe® Reader®. These applications are very straightforward and normally small in size (usually under 100MB). Very little, if any, modifications are needed to run these applications. | **Typical sequencing time: less than 1 hour** |
| Moderate | This is probably the most common application type. Moderate application types might require some changes while sequencing to function correctly, or may require no changes but have a larger install that takes more time. In rare instances, both scenarios can occur. Changes that might be encountered in these packages include making changes to the registry, altering the DeploymentConfig or UserConfig file to launch with additional parameters and scripts, or there may be additional applications needed to install together as a suite to allow cross-functionality. | **Typical sequencing time: 1-4 hours** |
| Complex | These are large applications or applications that take four or more hours to install, significant amounts of customization to function in the virtual environment, or both. Packages like these will normally be larger than 1 GB in size and take an extended period of time to sequence. Other hurdles that may be encountered are the application’s reliance on files being in a specific place and functions hard-coded to the install. These applications may require manually editing batch and command files to point to resources in the virtual environment. If this is the case, it is highly recommended to utilize a program that can scan multiple files and make several changes at once. It may also be required to install a device driver separately since drivers cannot be virtualized. Applications of this complexity can be sequenced; however it is imperative that all the pieces are in place before beginning the sequencing process. All knowledgeable resources should be engaged and available, sequencing hardware should be better than average, and finally, sequencing applications such as these should be done by an experienced sequencer. | **Typical sequencing time: 4-8 hours, but could be longer depending on the size and number of files** |

Table : Classifications of Applications

# Sequencing Limitations

Sometimes there are applications that cannot or should not be sequenced. Also there are certain limitations with App-V. Here is a list of application functions and limitations of App-V sequencing.

|  |  |
| --- | --- |
| Limitation | Description |
| Applications that start services at boot time | * App-V requires a logged in user to initiate the launch of an application. |
| Applications that require device drivers | * App-V cannot virtualize drivers. It is possible to bypass this issue and install the driver locally on the target computer. * Some user-mode device drivers can be virtualized. |
| Applications that are required by several applications for information or access | * For example, a program that will initiate a command and launch another program. Normally both programs would be included in the same suite. However if this application launches or initiates commands in several applications it may not be feasible to include all of the applications in the same suite. * This is especially true if one of the reasons for deploying App-V is to avoid application conflicts. Always remember that the virtual “bubble” can see the OS and what’s installed on it but the OS cannot see the “bubble” and interact with it. On the same note, remember that one “bubble” cannot see another unless they are brought into the same virtual environment using Connection Groups. |
| Applications that are a part of the OS | * Such as Internet Explorer® |
| Applications that use COM+ | * Because COM+ is dynamic and happens at runtime, there’s no way for the Sequencer to capture this information. |
| COM DLL surrogate virtualization | * For example, DLLs that run in Dllhost.exe |

Table : Sequencing Limitations

# Sequence files

App-V Packages created during the sequencing process are comprised of several files that are stored in a selected directory. The following is a list of those files and a description of each of them.

|  |  |
| --- | --- |
| **File** | **Description** |
| .APPV | The Virtual Application Package file containing all assets and state organized into stream able feature blocks. |
| .MSI | Executable deployment wrapper allowing the manual deployment of .APPV files or deployment via existing third party deployment platforms. |
| \_DeploymentConfig.XML | Used for customizing the default publishing parameters for all applications in a package. |
| \_UserConfig.XML | Used for customizing the publishing parameters directed to specific user groups for all applications in a package. |
| .CAB | *Optional:* Package Accelerator file used to automatically rebuild a previously sequenced virtual application package. |
| .APPVT | *Optional:* Sequencer Template file used to retain commonly re-used sequencer settings. |

Table : Sequencing Output Files

# Sequencing Walk Through

This section outlines the sequencing process. To begin, open the Application Virtualization Sequencer and select **Create a Package** from the **Welcome Screen**.

## Welcome Screen

The Welcome Screen is new in App-V 4.6 SP1 and provides quick access to the common tasks of creating and upgrading a package.

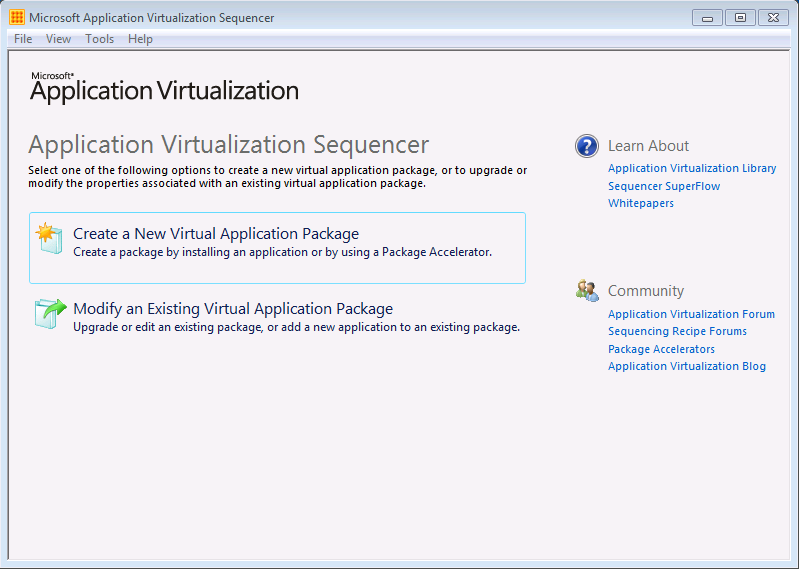


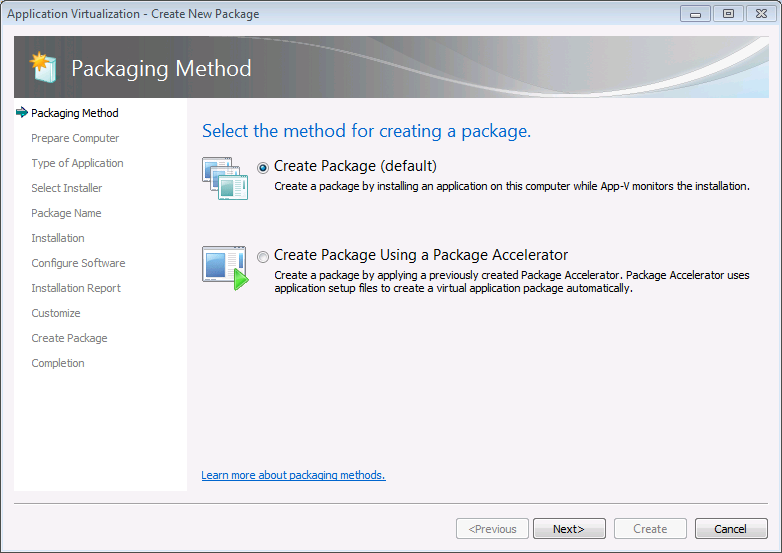
Figure : Welcome Screen

|  |  |
| --- | --- |
| Welcome Screen Component | Description |
| Create a New Virtual Application Package | This option will launch the wizard for creating a new virtual application package. |
| Modify an Existing Virtual Application Package | This option will launch the wizard for modifying or upgrading an existing virtual application package. |

Table : Welcome Screen Component Description

## Packaging Method

Creating a package starts with selecting which method to use to create the package. The picture and table below describe the different options that are available.



**Figure 2: Packaging Method**

|  |  |
| --- | --- |
| Package Method Component | Description |
| Create Package | Select this option to create a new package from scratch. |
| Create Package Using a Package Accelerator | Select this option to create a new package from a previously created Package Accelerator. |

Table : Package Method Component Description

## Prepare Computer

Next, the Sequencer examines the current operating environment to evaluate running processes or conditions that are in place (e.g. the Sequencer has not been reverted to a clean state after a previous sequencing operation, or there are pending reboot operations) that might prevent successful sequencing.

**Note:** The example below displays a Sequencer that has identified the Windows Defender service as active and recommends stopping the service before continuing.

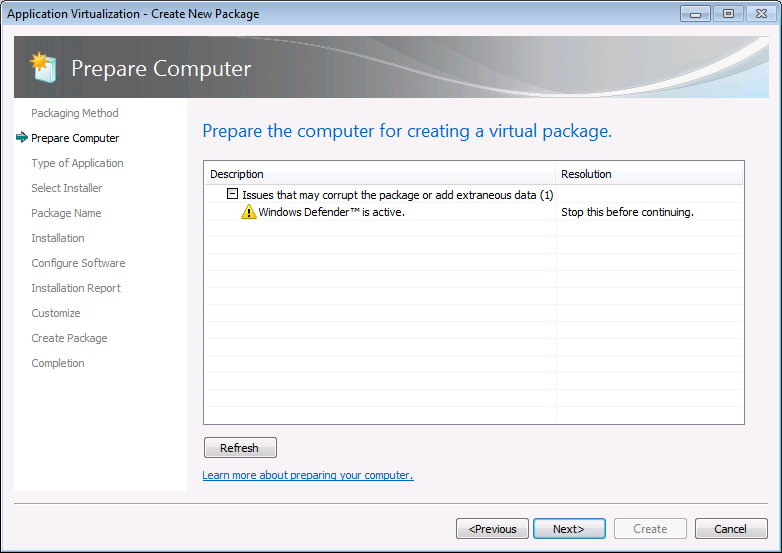


Figure : Prepare Computer

## Type of Application

The Sequencer offers choices for the type of application being sequenced, and provides tailored steps in the wizard, thereby simplifying and reducing errors during sequencing. Select **Standard Application** for packages that are self-contained (ex: one application or one suite of applications that are isolated from other software). Select the **Add-on or Plug-in or the Middleware** options when creating a package that needs to interoperate or when creating packages for [Connection](#_Dynamic_Suite_Composition) Groups solutions.

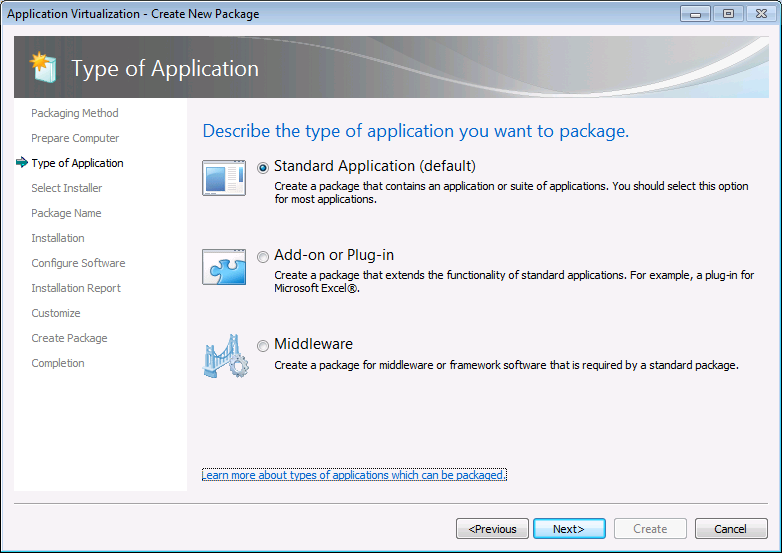


Figure : Type of Application

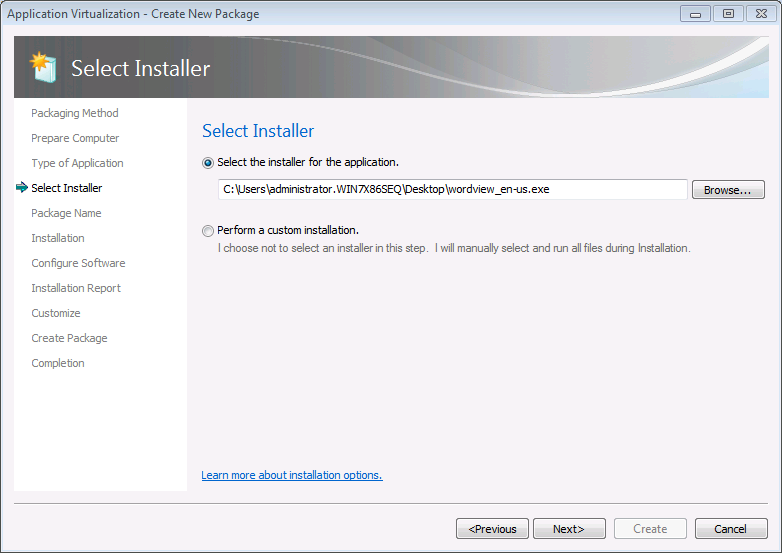
|  |  |
| --- | --- |
| Advanced Options Component | Description |
| Standard Application | Select this option when sequencing a single application or suiting multiple applications into the same virtual application package. |
| Add-on or Plug-in | Select this option when sequencing multiple applications in separate virtual application packages and linking them using a Connection Group. This option can also be used when packaging Add-ons or Plug-ins for locally installed applications like Internet Explorer. |
| Middleware | Select this option when sequencing multiple applications in separate virtual application packages and linking those using Connection Groups. This option will first create the application package for the middleware component and then create the second virtual application package that will contain the primary application. |

Table : Type of Application Component Descriptions

## Select Installer

Point the Sequencer to the installer for the application being sequenced. An “installer” can be any executable file designed to install the desired application. The Sequencer will automatically launch the installer when it activates monitoring.

Alternatively, “Perform a custom installation” can be selected. This option causes the sequencing wizard to enter monitoring and then wait for manual launching installation tasks. This option is often useful when sequencing applications that may not have an install or setup file such as applications that copy from a network share.

**Figure 5: Select Installer**

|  |  |
| --- | --- |
| Advanced Options Component | Description |
| Select the installer for the application | This option typically applies to sequencing only a single application using a single installer. Select this option and define the installer file to have the sequencing wizard automatically launch the installer. |
| Perform a custom installation | Select this option to start monitoring and manually launch application installer(s). This option behaves similar to previous versions of the Sequencer. It is useful when sequencing with multiple application installers or when there is no application installer. |

Table : Select Installer Component Descriptions

## Package Name

Select a package name, typically something descriptive of the vendor, software and version. This name is independent of the Primary Virtual Application Directory, but should be noted for saving the package. Saving the package in a directory named for the package name is recommended. At the package name screen, select the Primary Virtual Application Directory.

The Primary Virtual Application Directory is the directory that will contain all files for the sequence. It is recommended to define the application’s default installation directory (example C:\Program Files\directory) as the Primary Virtual Application Directory.

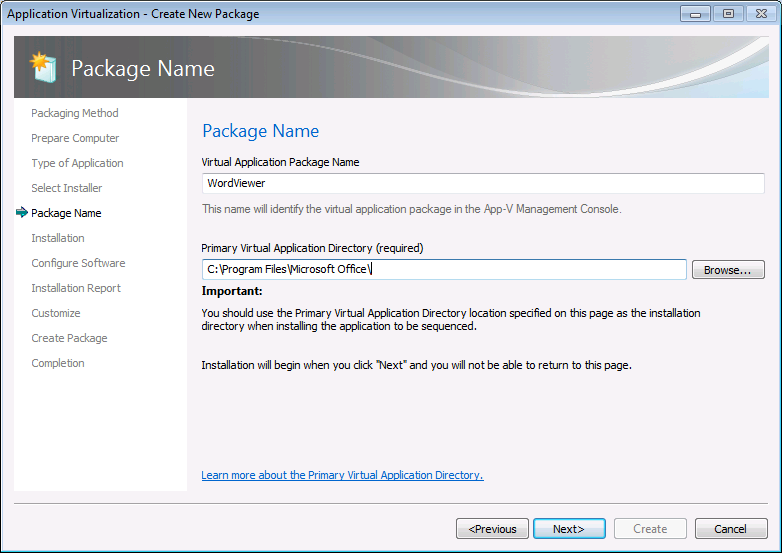


Figure : Package Name

|  |  |
| --- | --- |
| Advanced Options Component | Description |
| Virtual Application Package Name | Define a unique name for the virtual application package. |
| Primary Virtual Application Directory | The Primary Virtual Application Directory is the directory that will contain all files for the sequence. It is recommended to define the application’s default installation directory (example C:\Program Files\*directory*). |

Table : Package Name Component Description

## Installation

Install the application using the steps documented when locally installing the application. The Sequencer will begin monitoring and will automatically launch the installer defined earlier. If manual installation is required, the application installer can be directly launched, or click **Run** and select an installer to be launched by the Sequencer. When multiple installers are required to create the package, click **Run** after the completion of each installer, select the next installation program, or manually launch the installer until all installers have been successfully installed.

Once all installations are complete, select the **I am finished installing** check box and click N**ext**.

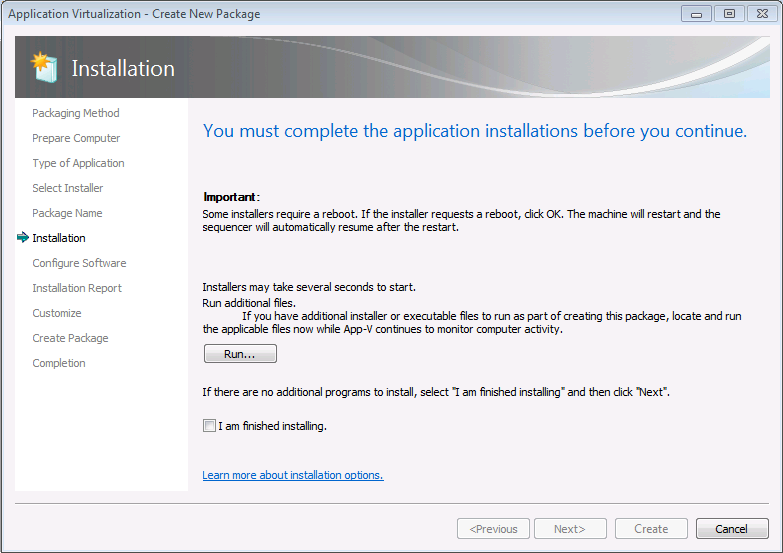


Figure : Installation

### Applications that Require a Reboot

For applications requiring a reboot, if given the option click “yes.” The Sequencer will reboot and automatically continue once logged back on to the sequencing workstation after the reboot has taken place.

## Configure Software

Many applications have first-run tasks such as accepting license agreements, etc. At this stage, execute the application(s) at least once by selecting the application and clicking the “Run Selected” or “Run All” buttons (multiple executions are recommended to ensure any second-run tasks are executed). Also, it is during this execution that any applicable application configuration changes should be made.

**Note:** This screen is also running in monitoring mode. It is possible to manage the tasks for programs that are not listed on this page by launching them outside of the Sequencer using Windows Explorer.

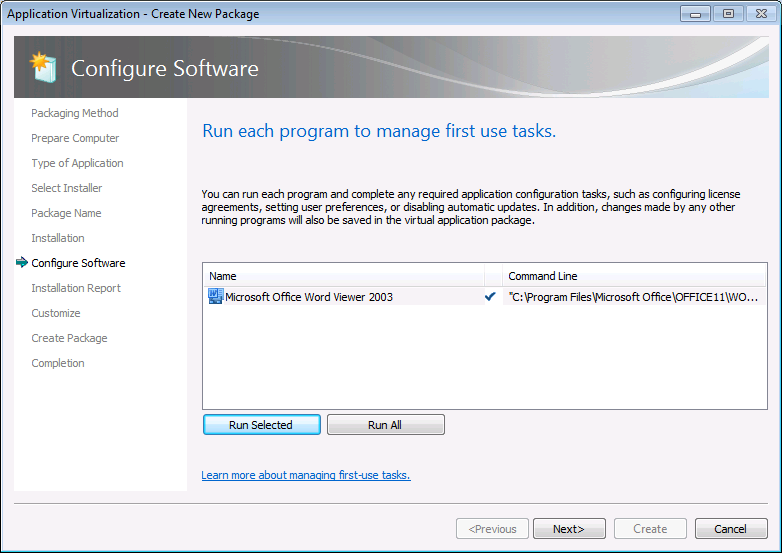


Figure : Configure Software

## Installation Report

The Sequencer detects common issues during sequencing. The Installation Report page of the wizard displays diagnostic messages categorized into Errors, Warnings, and Info depending on the severity of the issue. Double click an item in the report to view detailed information about the issue as well as suggestions for resolution. Messages from the system preparation report as well as the installation report are summarized upon package completion and are saved along with the package in a report.xml file.

Items in the report include:

* Excluded Files
* Drivers
* COM+ System differences
* SxS Conflicts
* Shell Extensions
* Files or registry entries that were not captured during monitoring

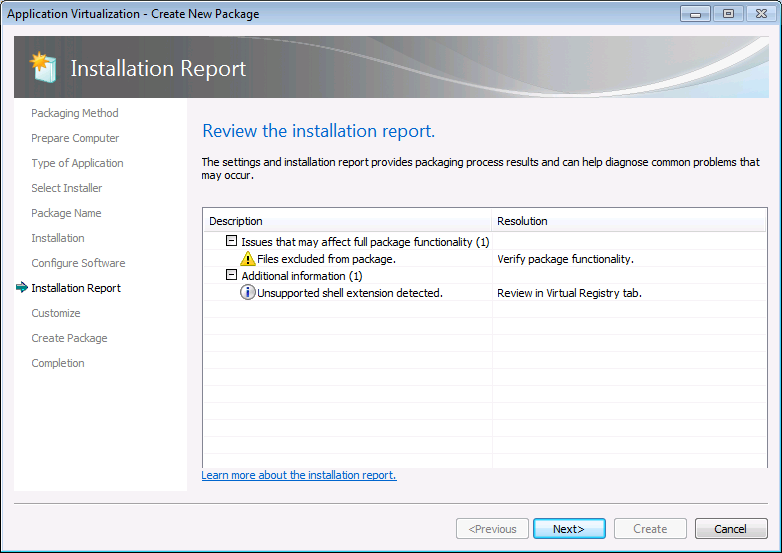


Figure : Installation Report

## Customize

Choose **Stop now** if the sequence will not benefit from further customization and select **Create**. However, often there are other steps remaining such as:

* Splitting the package into feature blocks to reduce the streaming requirement and save bandwidth.
* Selecting additional client operating systems that will be permitted to receive this package.
* Changing shortcuts and file type associations.
* Modifying registry settings and adding and deleting files in the package.

When additional customization is required select **Customize** and **Next** to continue the sequencing process and allow additional changes prior to the creation of the package.

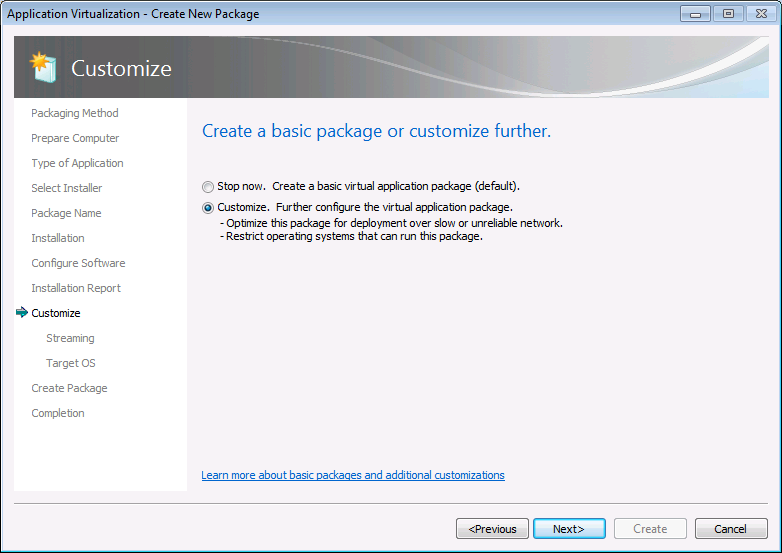


Figure : Customize

|  |  |
| --- | --- |
| Advanced Options Component | Description |
| Stop now | Select this option when confident that the application as is requires no additional configuration, and does not require the use of Feature Blocks. |
| Customize | This option will continue the sequencing process and:   * Create Feature Blocks to reduce the amount of traffic used for deployment * Define which operating systems may and may not run the package |

Table : Customize Component Descriptions

## Prepare for Streaming

Feature blocks are designed to optimize the applications for streaming (if applicable), creating a minimum launch threshold that allows launching larger applications as soon as enough of the package has been downloaded and does not require downloading the entire package. This enables users access to applications more quickly upon deployment.

Feature blocks also reduce the total network bandwidth used when launching the application for the first time and saves hard disk space on the client by leaving less-used data on the server until it is specifically called by the user.

Creating feature blocks is recommended unless the deployment method for virtual application packages is only completed with System Center Configuration Manager for Download Locally and Run option or via MSI for standalone mode clients.

At the Prepare for Streaming screen, feature blocks are created based on individual, selection of, or all applications.

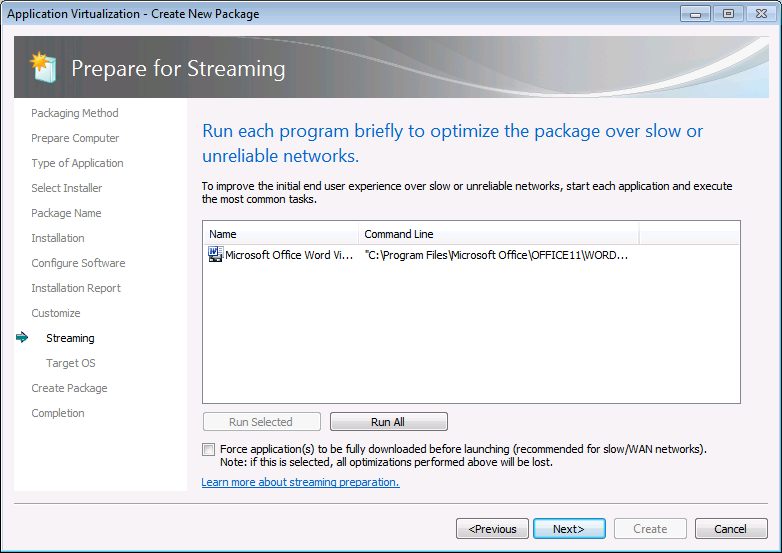


Figure : Prepare for Streaming

|  |  |
| --- | --- |
| Launch Applications Component | Description |
| Run Selected | This will launch an application to ensure functionality. Additionally, any files that are executed during this phase will be tagged as Primary Feature Block. Any remaining files will be tagged as Secondary Feature Block. If no applications are launched then all files will be a part of the Primary Feature Block. |
| Run All | Performs the same operation as the launch component, but launches all identified applications in the package. This is useful when trying to create Primary and Secondary Feature Blocks for a large suite of applications. |
| Force download before launch | If enabled this option forces the client to download the entire package before launching the application. This may be useful for packages that will be downloaded over slow WAN links where application performance may be impacted. |

Table : Prepare for Streaming Component Descriptions

Select and run each shortcut from the package that users execute in typical day-to-day operations. Then, perform the common tasks that typical users perform within each particular application during normal operations.

During this process, the Sequencer tracks which specific pieces of the package’s resources are being executed and includes them in the primary feature block. When a user launches the application for the first time, the App-V client will stream and cache just the data within the primary feature block over the network and will launch the application.

Any pieces of the package not included in the primary feature block are placed in the secondary feature block and reside on the server or storage location until specific resources from within the secondary feature block are called by the App-V client. Those pieces are streamed on-demand and cached on the client.

Clicking ‘Next’ without launching any shortcuts enables the entire content of the package streaming and cached “on-demand” on the App-V client. Typically, this is done for very small application packages if streaming the entire package does not cause any network bandwidth concerns.

Normally, the client launches the application after the primary feature block has been downloaded to improve launch time. By selecting the “Force application(s) to be fully downloaded before launching” option, the client will be forced to wait until all blocks of the virtual application package have been downloaded before launching the application. This is useful when clients may be running this package over slow WAN links.

## Target OS

The Target OS screen configures the package targeting for specific operating systems and even specific platform architectures. For a packaged application that does not run on 64-bit operating systems, the 64-bit operating system targets must be deselected. This will ensure that the application will never be delivered to those machines. The same could be accomplished for specific Operating Systems where there is a compatibility issue.

Applications sequenced on a 32-bit OS are often likely to run on a 64-bit client OS. However, applications sequenced on a 64-bit OS will never run on a 32-bit client OS. Certain applications that are compatible with both 32-bit and 64-bit operating systems may only work if sequenced separately on a 32-bit Sequencer and a 64-bit Sequencer. This screen enables targeting two separate packages of the same application to the appropriate client operating systems.

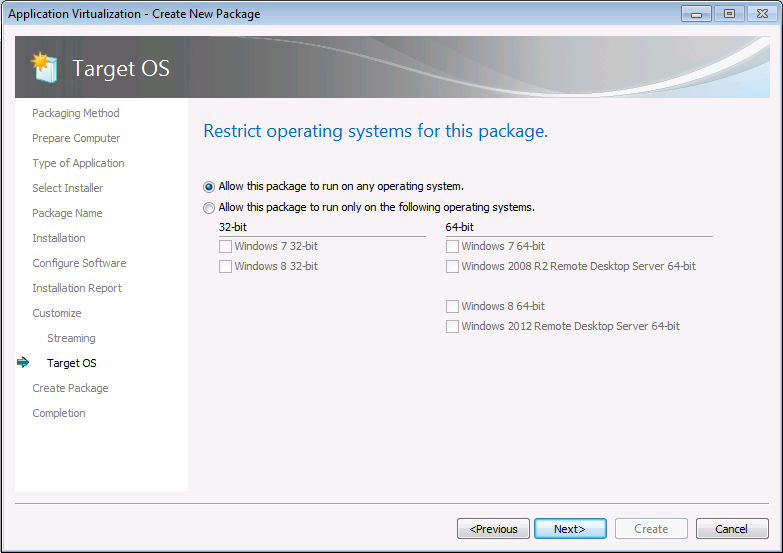


Figure : Target OS

## Create Package

After completing the streaming and target OS selection, the package can be saved. If the package is saved from this step the sequencer can place a description of the process into the package as well as the save location for the package. Specify a save location directory outside the source files and utilize the package name as the directory (Example: Desktop\WordViewer).

Choosing “Continue to modify package without saving using the package editor” and clicking **Next** will progress to the Completion report then exit to the Package Editor allowing for the configuration of shortcuts and file type associations, registry values or services.

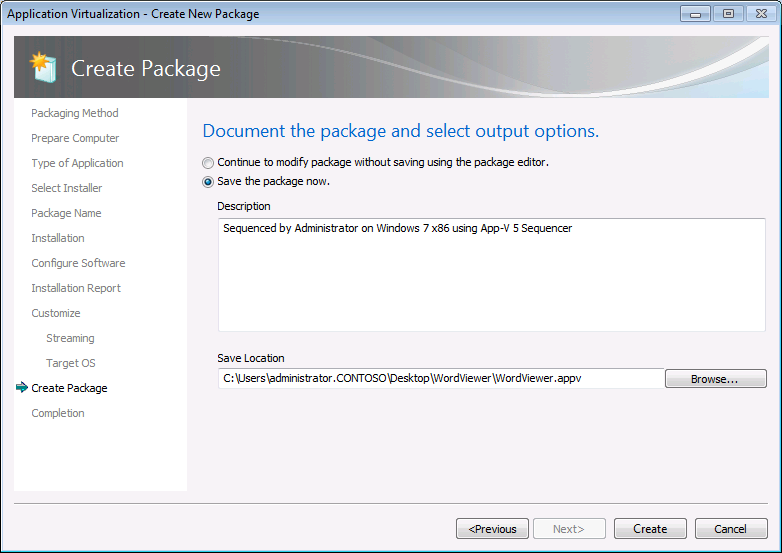


Figure : Create Package

## Completion Report

The Completion Report page of the wizard displays diagnostic messages categorized into Errors, Warnings, and Info depending on the severity of the issue. Double click an item in the report to view detailed information about the issue as well as suggestions for resolution. Messages from the system preparation report as well as the installation report are summarized upon completing the package and are saved along with the package in a report.xml file.

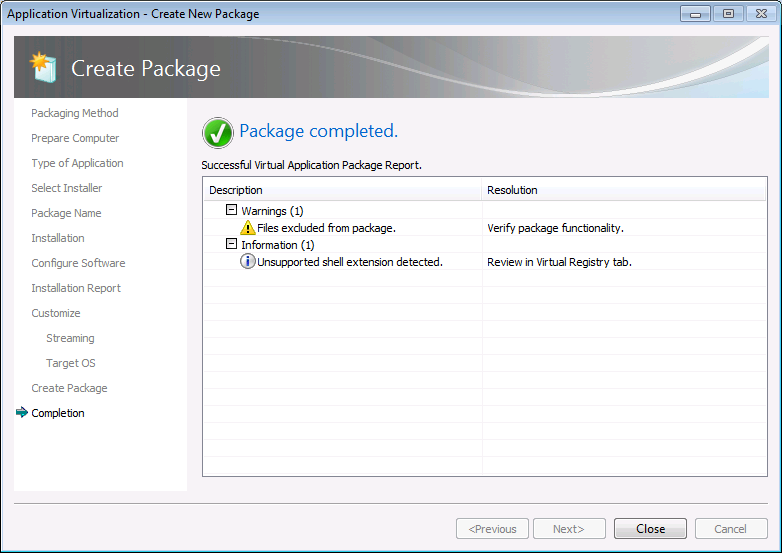


Figure : Package Completed

## Package Editor

The package editor is comprised of several tabs that enable further configuration modification prior to saving the package. These tabs include options to modify the following settings:

* Changing shortcuts and file type associations.
* Modifying registry settings.
* Adding and deleting files in the package.

The **properties** tab displays the current properties of the package and allows for changing of the Package Name and setting a Description.

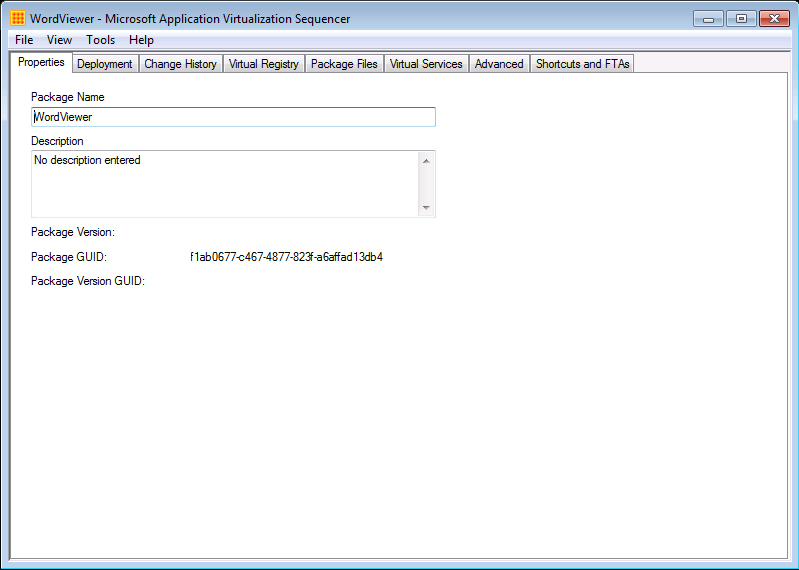


Figure 17: The Properties Tab

**The Deployment tab** displays the current OS configuration and allows the OS configuration settings modification.

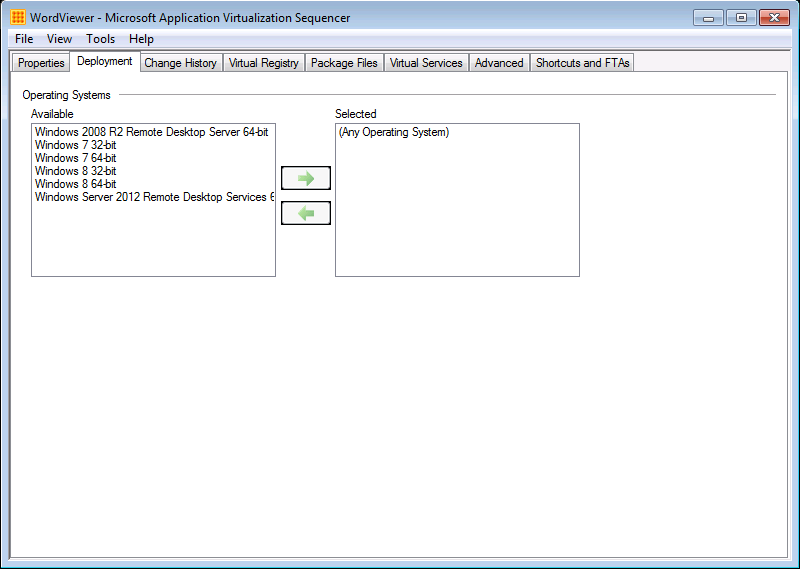


Figure 18: The Deployment Tab

**The Virtual Registry tab** displays the current virtual registry configuration and allows for deleting or renaming existing keys and values as well as adding new keys and values in both the HKLM (Machine) and HKUSERS (Users) hives.

Where the same registry key may exist on the local system as well as in the virtual application package, the virtual key can be configured to either merge with the local configuration or override the local configuration.

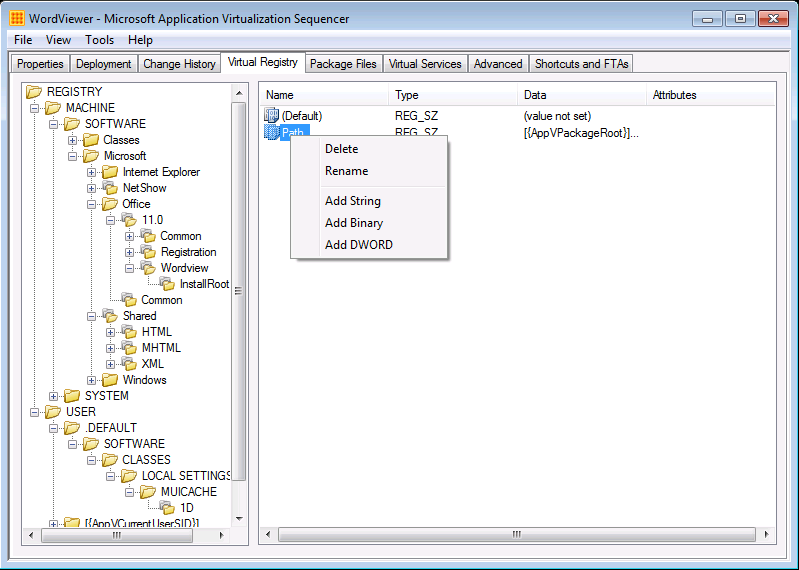


Figure 19: The Virtual Registry Tab

The **Package Files** tab displays the current list of files and folders added to the package and allows for the addition or deletion of files. However, this interface should not be used to add or remove files in the package if the package has previously been optimized for streaming by way of creating feature blocks.

* **Override Local Directory:**  If the same folder or file exists outside the package as well as inside the package, this option instructs the client to ignore the local file or folder and only view the file or folder inside the package.
* **Merge with Local Directory** If that the same folder exists outside the package as well as inside the package this option instructs the client to view the contents of both the virtual and local folders.

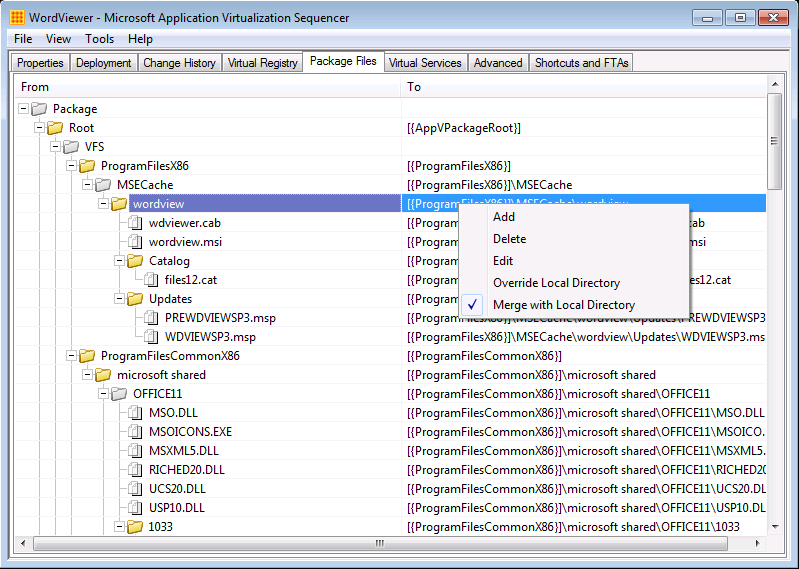


Figure 20: The Package Files Tab

The **Services** tab displays the current configuration of virtualized application services and allows for changing the Startup Type, Log On and Dependencies configuration of virtualized services.

**Note:** The services tab is read-only. In order to disable a virtual service, set the service’s properties during monitoring using Services.MSC or by utilizing a dynamic configuration file post-sequencing, as described later in this document.

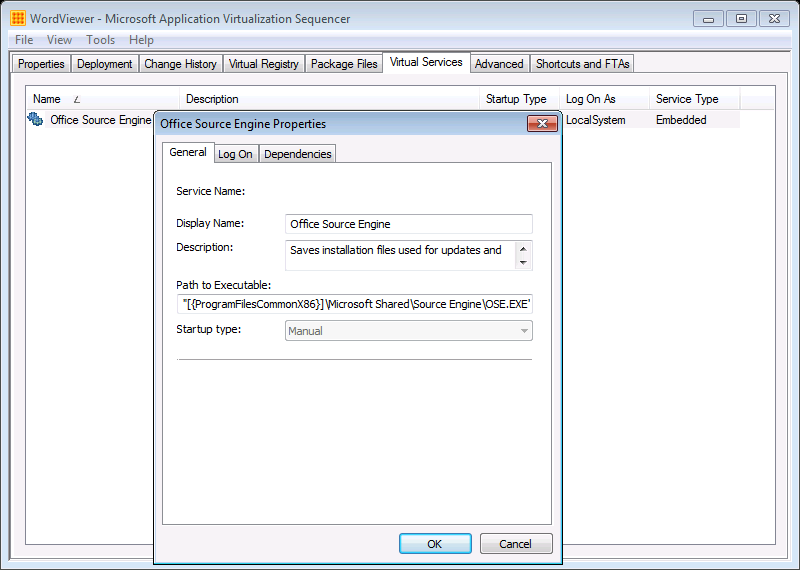


Figure 21: The Services Tab

The **Advanced** tab provides an option to enable visibility of named and COM objects in an App-V package to the local system to improve the usability of some application functions. Local system visibility may be useful for such tasks as virtualizing legacy versions of Microsoft Outlook®.

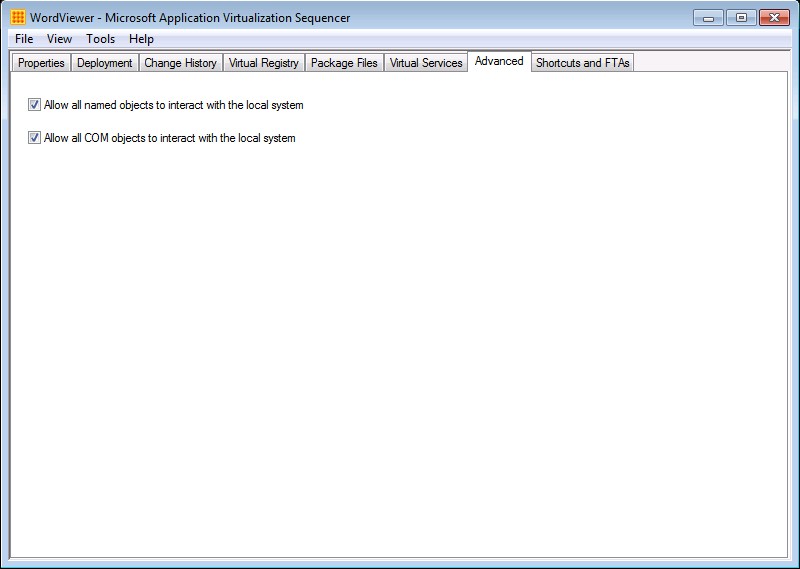


Figure 22: The Advanced Tab

The **Shortcuts and FTAs** tab provides the ability to customize the Shortcuts and File Type Associations for the applications identified during monitoring. Applications may have to be added or removed from this list, based requirements for the final package. In addition, with web-based applications it is often necessary to add Internet Explorer as an application where the web-based application requires launching Internet Explorer as a dependency.

Each application can be modified to change the name, icon, file type associations, and locations for shortcuts on destination computers.

* To **add a new application shortcut** right-click the **Applications** node in the left hand tree and launch the **Add Application** window to define the application path, name and version for the shortcut.
* To **edit an existing shortcut** right-click the desired application under the **Applications** node in the **Edit Applications** window, change the path, name and version of an existing shortcut.
* To **edit the location** of an existing shortcut, expand the desired application in the tree and right click the **Shortcuts** node and select **Edit Locations. T**he **Shortcut Locations** window shown in Figure 23 allows for changing shortcut placement.

Changes to application **File Type Associations** are done in the same location with a similar interface.

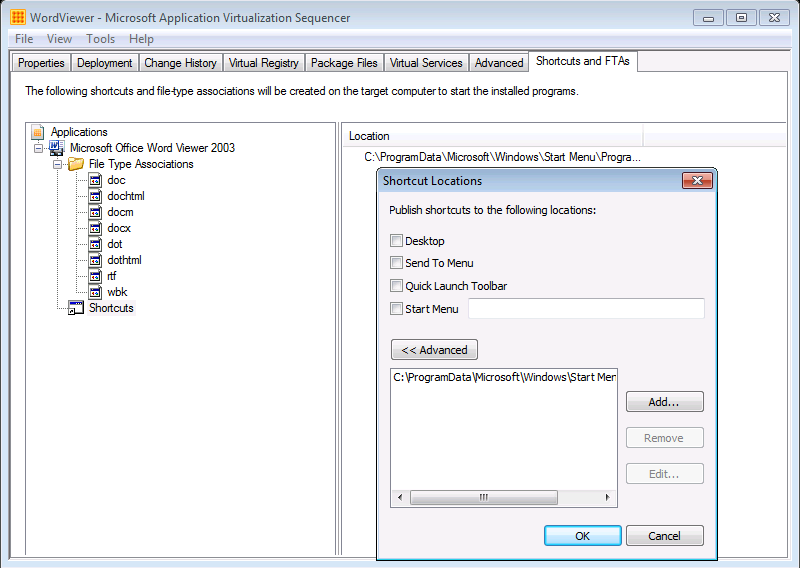


Figure 23: The Shortcuts and FTAs Tab

**Save As**

When finished making customizations, select the **File** pull-down menu and select **Save** or **Save As** to save the virtual application package.

As a recommended best practice, create a new folder for the package using the Package Name and save the package in this folder. Once saved, copy the package folder to a preferred package repository.

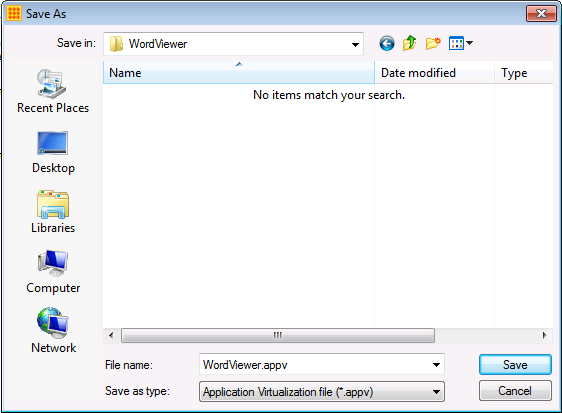


Figure 24: SaveAs

## Sequencing with PowerShell

The App-V v5 Sequencer allows sequencing of App-V packages outside of the Sequencer UI using only PowerShell commands.

All sequencer functions must be performed in an **elevated** PowerShell console and the PowerShell module for the sequencer must be loaded. To load the Sequencer PowerShell module execute the following command at the PowerShell prompt: **Import-Module AppVSequencer.**

Sequencing via the PowerShell cmdlets provide similar options to those available when sequencing from the GUI. Utilizing PowerShell for sequencing applications provides automation of transitioning from MSI formatted packages to App-V. The powerful interface gives the capability creating packages without walking through wizards for well-defined applications.

Example command syntax for performing a sequence of Microsoft Orca is provided in this section. This example assumes the following:

* The name of the new virtual application package is to be **Microsoft Orca** (specified in the **–Name** option).
* The installation file for Microsoft Orca (Orca45.MSI) has been placed locally on the sequencing workstation in C:\SourceFiles (specified in the **–Installer** option).
* The destination folder will be created by the sequencer within C:\Packages (specified in the **–Path** option). The sequencer will create a new folder for the package using the package name within the folder specified in the **–Path** option.

**Note:** The folder specified in the **–Path** option must already exist.

* The Primary Virtual Application Directory will be C:\Program Files\Microsoft Orca (specified in the **–PrimaryVirtualApplicationDirectory** option).

The PowerShell command syntax for sequencing Microsoft Orca is below:

**New-AppvSequencerPackage –Name “Microsoft Orca” –Path “C:\Packages” –Installer “C:\SourceFiles\Orca45.msi” –PrimaryVirtualApplicationDirectory “C:\Program Files\Microsoft Orca”**

The above command string will instruct the sequencer to create a new App-V package called “Microsoft Orca” from the “Orca45.msi” file defined in the **–Installer** option using “C:\Program Files\Microsoft Orca” as the **–PrimaryVirtualApplicationDirectory.** The sequencer will save the newly created package files in a sub-folder named as the package name in the “C:\Packages” folder defined in the **–Path** option of the string.

For further details and additional options available when using the **New-AppvSequencerPackage** cmdlet, issue the following command from PowerShell:

**Get-Help New-AppvSequencerPackage**

# Creating a Package Accelerator

After a package has been sequenced, a Package Accelerator can be created for that package. App-V Package Accelerators enable application packaging easily and automatically without monitoring the application installation while following a step-by-step recipe.

The Sequencer wizards guide packaging engineers through the package creation wizard when utilizing a package accelerator. Behind the scenes, the Sequencer creates a new package from the App-V Package Accelerator and installation media without monitoring the installation of the application. As with sequencing applications without a package accelerator, the sequencing workstation should be reverted after applying an App-V Package Accelerator. The Sequencer also has a wizard for the creation of an App-V Package Accelerator.

A Package Accelerator (*packagename*.CAB) file will contain the following types of files:

* DeploymentConfig and UserConfig XML files.
* Manifest file
* Any files created by the app at time of sequencing containing customizations etc. that are not provided by the installation files
* Files necessary for the virtual environment and streaming optimization

Creating an App-V Package Accelerator requires the App-V Package and installation files for the application or the location of the application files that have already been locally installed.

Supported Installation Media file types can be presented to the Sequencer for creating or using Package Accelerators:

* .MSI
* .ZIP
* .CAB
* Individual files and folders
* **NOTE:** Self-extracting executable files are not supported. If using this type, first extract the files and present the folder containing the extracted set-up files to the sequencer.
  + Alternatively, at the Installation Files Screen (Figure 26) select the option ‘Files installed on local system’. To use this, the application must be already physically installed on the local system.

## Work-flow

* In the Package Editor, after a sequence has been completed, click the “Tools” pull-down menu and select “Create Accelerator.”

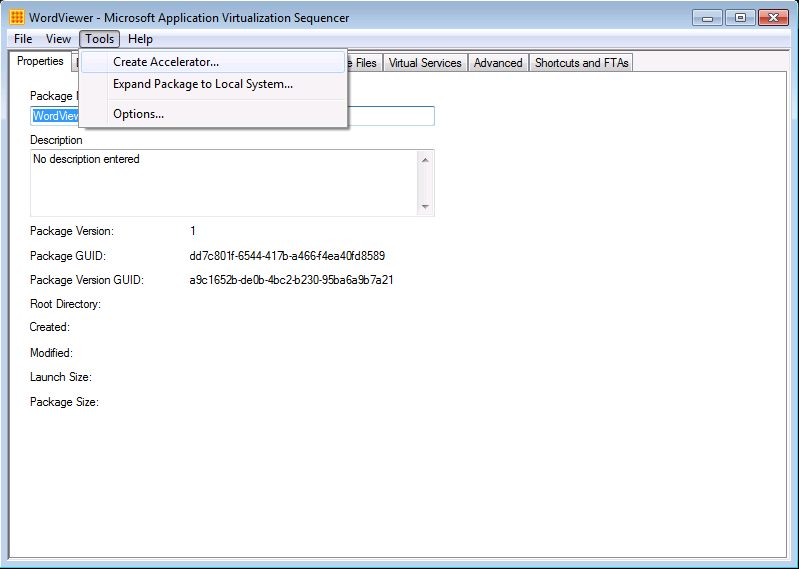


Figure 25: Tools Menu

* At the **Select Package** screen, browse to and select the **APPV** file for the package to use in order to create the Package Accelerator.
* At the **Installation Files** screen, specify the location of the application’s installation files.
  + Select **Original installation files** and browse to and select a root folder that contains all installation files for the application

**OR**

* + Select **Files installed on local system** and browse to and select the folder to which the application was already installed. (Must be installed to its default location).

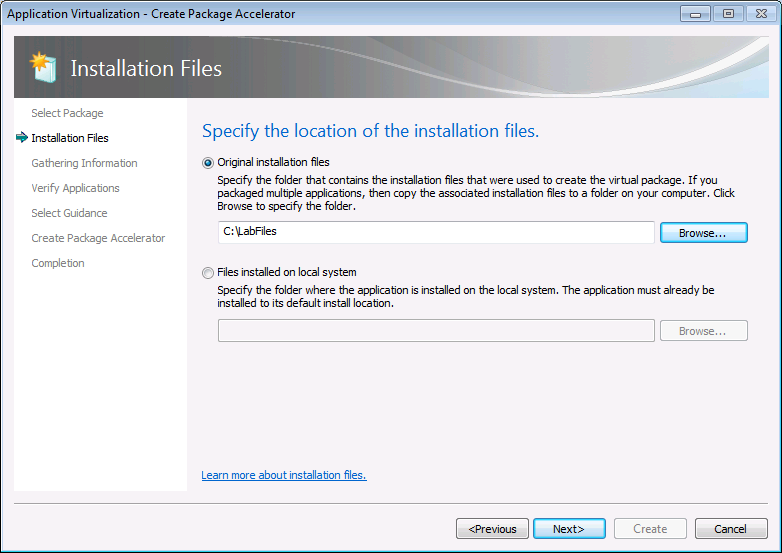


Figure 26: Installation Files

* At the **Select Files** screen, the Sequencer will display all files detected for use in the App-V Package Accelerator. Uncheck the boxes next to any unwanted file(s) to exclude them from the Package Accelerator.
  + Example: The Sequencer has included a large temporary or cache location containing copies of the setup files not needed in the package. Removing files such as this can help reduce the size of the overall package.

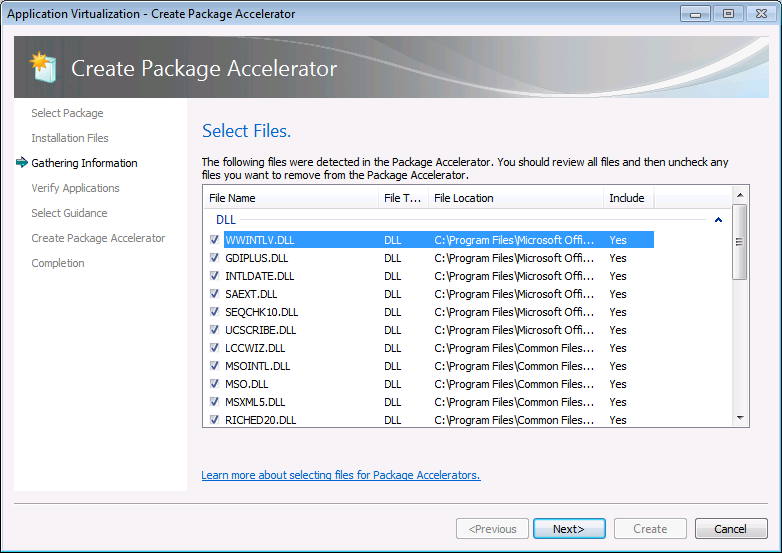


Figure 28: Select Files

* At the **Verify Applications** screen, the Sequencer will provide a list of applications detected in the package for reference purposes for the engineer who will later create the package from the accelerator. This information is useful for the engineer to know which application installers will be needed.

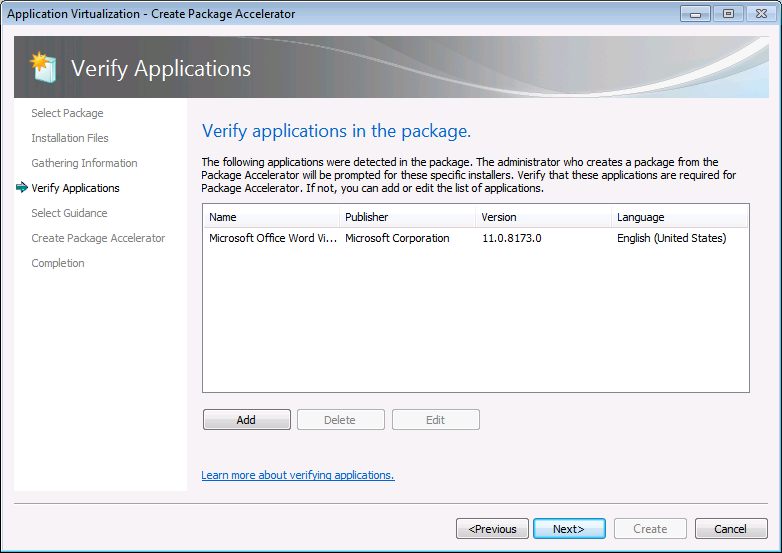


Figure : Verify Applications

* At the **Select Guidance** screen, select an RTF file containing any special instructions needed for the engineer to create the package using the App-V Package Accelerator. The App-V Package Accelerator will then display the information to the engineer performing the sequencing later.

Some examples of information to include are:

* + Name of original packager for internal reference (if applicable)
  + Names and versions of the specific applications included in the package
  + Supported platforms
  + Prerequisites
  + Special instructions or configurations
* At the **Create a Package Accelerator** screen, select a location to save the Package Accelerator. The Package Accelerator consists of a single CAB file.
* Save this file to the location of the installation media for easy retrieval.

## Creating a Package Accelerator with PowerShell

The App-V v5 Sequencer provides the capability to create App-V package accelerators outside of the Sequencer UI using only PowerShell commands.

All sequencer functions must be performed in an **elevated** PowerShell console and the PowerShell module for the sequencer must be loaded. To load the Sequencer PowerShell module execute the following command at the PowerShell prompt: **Import-Module AppVSequencer.**

After an App-V package has been created, a Package Accelerator for that package can be created.

There are two different command strings that can be used to create the package accelerator depending on the location of the source files (source directory or locally installed).

Example command syntax for creating a Package Accelerator for the Microsoft Orca package is provided in this section. This example assumes the following:

* The application files that were installed during the previous sequencing of Microsoft Orca exist in C:\Program Files\Microsoft Orca (specified in the **–InstalledFiles** option).
* The location of the Microsoft Orca virtual application package is C:\Packages\Microsoft Orca\Microsoft Orca.appv (specified in the **–InputPackagePath** option).
* The folder to which the sequencer will output the Package Accelerator file is C:\Packages\Microsoft Orca (specified in the **–Path** option).
* The location of the Accelerator Description or guidance file is C:\SourceFiles\guidance.rtf (specified in the **–AcceleratorDescriptionFilePath** option).

The PowerShell command syntax for creating the Package Accelerator is below:

**New-AppvPackageAccelerator –InstalledFilesPath “C:\Program Files\Microsoft Orca” –InputPackagePath “C:\Packages\Microsoft Orca\Microsoft Orca.appv” –Path “C:\Packages\Microsoft Orca” –AcceleratorDescriptionFilePath “C:\SourceFiles\guidance.rtf”**

When Creating a Package Accelerator with PowerShell from source files rather than with installed program files:

Replace: **-InstalledFilesPath**

With: **-Installer “*path to installer file”*** (Example: C:\SourceFiles\Orca45.MSI)

For further details about using the **New-AppvPackageAccelerator** cmdlet, issue the following command from PowerShell:

**Get-Help New-AppvPackageAccelerator**

# Create a package using a Package Accelerator

When creating a new virtual application package using an App-V Package Accelerator, first install the application locally on the sequencing workstation, or the installation files should be copied to the sequencing workstation. The Package Accelerator (.CAB) file should also be copied to the sequencing workstation.

## Work-flow

1. Select **Create a New Virtual Application Package** from the welcome screen.
2. At the **Packaging Method** screen, select **Create a Package using a Package Accelerator.**
3. At the **Select Package Accelerator** screen, browse to and select the relevant Package Accelerator CAB file.
4. At the **Guidance** screen, read carefully any important information that may be presented from the developer of the Package Accelerator. This information can be useful in creating a successful package.
5. At the **Select Installation Files** screen, browse to and select the root folder containing the installation files for the application.
6. At the **Package Name** screen, enter a name for the package. By default, the Sequencer will automatically enter the name of the original package that was used to create the Package Accelerator.
7. At the **Save the Virtual Application** screen, enter any comments relevant to any customizations that will be made to the package (optional) and select the location to save the new package, as well as the option whether or not to compress the package.
8. At the **Configure Software** screen, select the **Configure Software** button to proceed with launching the applications to provide any additional configurations before saving the package.
   1. If uncertain whether any further configuration is necessary, select to **Skip this step** and proceed directly to saving the package.
9. At the **Run Each Program to manage first-use tasks** screen, run each application at least once to clear any “first use” pop-ups and to make any additional configuration changes within the applications.
10. Continue to save the application.
11. Just as earlier in Section 8, upon completion the Sequencer will provide a report detailing any relevant information to the success of the package.

## Create a package using a Package Accelerator with PowerShell

The App-V 5 Sequencer provides the capability to sequence App-V packages from package accelerators outside of the Sequencer UI using only PowerShell commands.

All sequencer functions must be performed in an **elevated** PowerShell console and the PowerShell module for the sequencer must be loaded. To load the Sequencer PowerShell module, execute the following command at the PowerShell prompt: **Import-Module AppVSequencer.**

Two different command strings can be used to create the package from a Package Accelerator depending on the location of the source files (source directory or locally installed).

Example command syntax for creating a package for Microsoft Orca from a Package Accelerator is provided in this section. This example assumes the following:

* The name of the package will be Microsoft Orca (specified in the **–Name** option).
* The destination folder will be created by the sequencer within C:\Packages (specified in the **–Path** option). The sequencer will create a new folder for the package using the package name within the folder specified in the **–Path** option.

**Note:** The folder specified in the **–Path** option must already exist.

* The location of the Package Accelerator for Microsoft Orca is C:\SourceFiles\Microsoft Orca.CAB (specified in the **–AcceleratorFilePath** option).
* The application files that were installed during a previous local install of Microsoft Orca exist in C:\Program Files\Microsoft Orca (specified in the **–InstalledFiles** option).

The PowerShell command syntax for creating the package is as follows:

**New-AppvSequencerPackage –Name “Microsoft Orca” –Path “C:\Packages” –AcceleratorFilePath “C:\SourceFiles\Microsoft Orca.cab” –InstalledFilesPath “C:\Program Files\Microsoft Orca”**

When creating a Package from a Package Accelerator with PowerShell from source files rather than locally installed files:

Replace: **-InstalledFilesPath**

With: **-InstallMediaPath “*path to installer file”*** (Example: C:\SourceFiles\Orca45.MSI)

For further details and additional options available when using the **New-AppvSequencerPackage** cmdlet, issue the following command from PowerShell:

**Get-Help New-AppvSequencerPackage**

# Application Package Upgrade

Throughout the lifecycle of an application, it will need to be updated. It is important to understand the options associated with updating an application, and when to use them.

Three different types of Application Upgrades are available at the Select Task screen (after clicking Modify an Existing Virtual Application Package in the main sequencer screen):

* Update Application in Existing Package
* Edit Package
* Add New Application

**Update Application in Existing Package** (Most Common)

Select this option to apply an update to an application or program that is part of an existing virtual application package. Selecting this option assumes the associated update installation files are saved locally to the App-V Sequencer. Updating an App-V package requires access to the package that is being updated.

The wizard will walk through:

* Extracting the existing virtual application package on to the Sequencer for updating
* Installing the application updates
* Saving the updated package version

**Edit Package**

Select this option to modify the properties and configuration associated with an existing virtual application package rather than adding or updating files. Editing a package requires access to the location where the virtual application package is saved. The following displays the list of package properties that can be updated utilizing Edit Package:

* View package properties.
* View package change history.
* View associated package files.
* Edit registry settings.
* Review additional package settings (except operating system file properties).
* Create associated Windows Installer (MSI).
* Modify OSD file.
* Compress and uncompress package.
* Set virtualized registry key state (override / merge).
* Set virtualized folder state.
* Edit virtual file system mappings.

**Add New Application**

Select this option to add a new application or program to an existing virtual application package. In this mode, the sequencer will also look for and add new shortcuts for added applications, such as adding an Office Add-in to an existing Microsoft Office® virtual application package. Adding a new application requires access to the location where the virtual application package is saved.

The wizard will walk through:

* Extracting the existing virtual application package on to the sequencer for updating
* Installing the additional applications
* Saving the updated package version

## Updating a Package that will Replace the Existing Package

Sometimes a package needs to be updated and will subsequently replace an existing package. Upgrading a package is a necessary part of the application lifecycle when there is a patch, service pack, or other update from the software manufacturer. In this case, follow these steps when sequencing and, subsequently, publishing in production:

* Copy the package to the Sequencer.
* Open the package for Package Upgrade by selecting “Modify an existing Virtual Application Package” from the Welcome Screen.
* Select the type of upgrade to perform at the Select Task screen.
* Run the installation wizard and make the appropriate changes to the package.
* After the installation, launch all shortcuts in the same manner as during the original sequence.
* Complete sequencing as usual.
* The Sequencer will save the updated package with a *\_#*at the end of the APPV file name. (example: WordViewer2007\_2.sft) The number corresponds with the version (iteration) of the application package. Apply this new version to the chosen distribution method (varies) to upgrade or replace the version currently deployed to the clients.

## Updating a Package for Deployment with the Existing Package

Sometimes, a package needs to be updated, but the business community still wants access to the existing package. This could be used to introduce a new pilot version of the application and, eventually, deprecate the previous version after proper testing. In this scenario, follow these steps:

* Copy the package to the Sequencer.
* Open the package using Modify and existing Virtual Application Package.
* Select the type of upgrade to perform at the Select Task screen.
* Run the installation wizard and make the appropriate changes to the package.
* Perform the steps in the sequencing wizard as usual.
* At the Create Package screen, select “Continue to modify package without saving using the package editor.”
* In the Shortcuts and FTAs tab in the Package Editor, edit the names and versions of the shortcuts to make them unique as two shortcuts can be published with the same name in the same location.
* Select the “File” pull-down menu then “Save As” to save the package (Figure 24). Check the “Save As New Package” checkbox to create a new GUID and a new package root.
* The Sequencer will save the updated package with the new name. This new package would be distributed alongside the previous version on the clients.

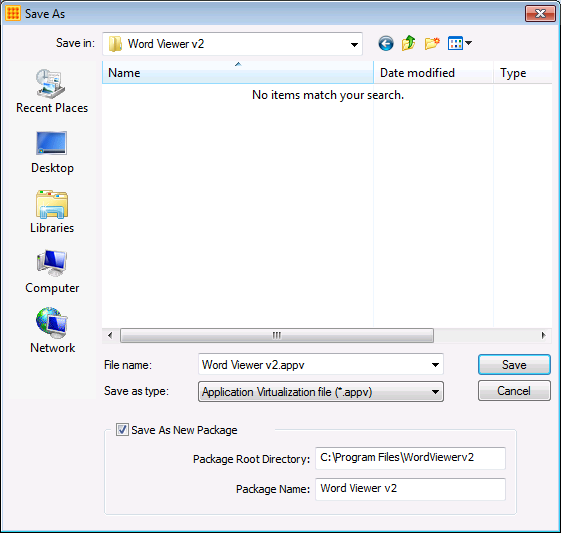


Figure 30: Save As New Package

## Update an existing package with PowerShell

The App-V v5 Sequencer also provides the capability to update App-V packages from outside of the Sequencer UI using only PowerShell commands.

All sequencer functions must be performed in an **elevated** PowerShell console and the PowerShell module for the sequencer must be loaded. To load the Sequencer PowerShell module execute the following command at the PowerShell prompt: **Import-Module AppVSequencer.**

Example command syntax for updating the Microsoft Excel® Viewer package is provided in this section. This example assumes the following:

* The path to the existing ExcelViewer package is C:\Packages\ExcelViewer\ExcelViewer.appv” (specified in the **–AppvPackageFilePath** option).
* The name of the package is ExcelViewer (specified in the **–Name** option).
* The destination folder in which to save the upgraded package is C:\Packages\ExcelViewer (specified in the **–Path** option).
* The installer file for the application upgrade has been placed in C:\SourceFiles\ExcelViewerSP2.MSI (specified in the **–Installer** option).

The PowerShell command syntax for upgrading the package is below:

**Update-AppvSequencerPackage –AppvPackageFilePath “C:\Packages\ExcelViewer\ExcelViewer.appv” –Name “ExcelViewer” –Path “C:\Packages\ExcelViewer” –Installer “C:\SourceFiles\ExcelViewerSP2.msi”**

There are additional options that can be used in this command syntax.

For further details and additional options available when using the **Update-AppvSequencerPackage** cmdlet, issue the following command from PowerShell:

**Get-Help Update-AppvSequencerPackage**

1. **Sequencing for Connection Groups (Plug-ins and Middleware)**

Connection Groups in App-V enables virtual applications to interact with other applications, middleware or plug-ins that have been virtualized in separate virtual application packages by sharing a common virtual environment. Applications virtualized in separate virtual application packages are very limited in the ways that they can interact with each other.

Connection Groups can be utilized when working with applications that depend on plug-ins such as ActiveX controls or for applications that depend on middleware such as OLE DB or the Java Runtime Environment (JRE). Sequencing the separate components reduces the servicing requirements. Sequencing the primary applications without dependent components and sequencing the middleware or plug-in as a secondary package enables individual servicing of the packages and allows the secondary packages to be reused with other primary applications.

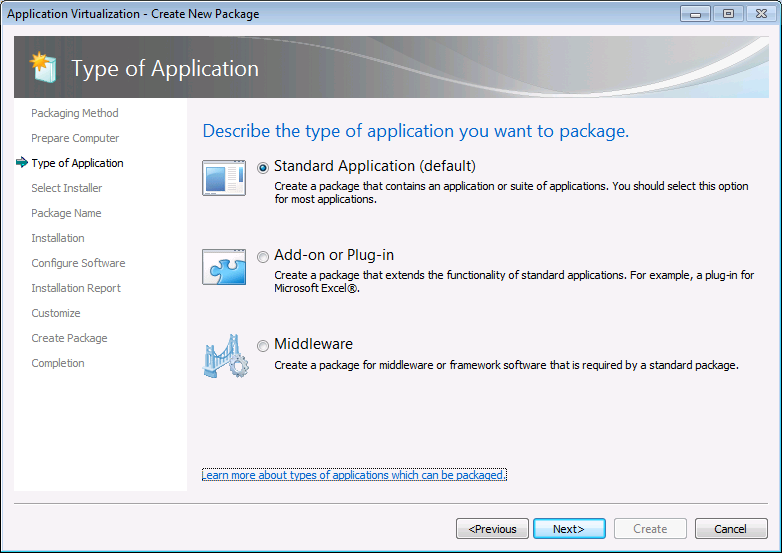
## Advantages

* Reduce the size of the primary packages.
* Provides better control of access permissions on the secondary applications.

## Workflow

The App-V 5 Sequencer user interface has been designed to include guided workflows for creating packages intended to be used in Connection Groups.

When selecting Create a New Virtual Application Package from the Welcome Screen, the wizard presents a workflow unique for the type of application selected. The “Add-on or Plug-In” and “Middleware” options are options that guide sequencing engineers through creation of packages that support connection groups.



**Figure 32: Type of Application**

## Add-on or Plug-in

When creating a separate package for a plug-in to the Primary Application, for example, a plug-in to Microsoft Excel, select the Add-on or Plug-in option.

The workflow for plug-ins is as follows:

1. Sequence Primary App (Type of Application: Standard Application (default)) and copy Package files to storage
   1. Follow the same process in **Section 8** of this document.
2. **Revert** the sequencing virtual machine to the base snapshot.
   1. **Copy** the primary app package back to the sequencing virtual machine.
3. Sequence the plug-in
   1. In the Sequencer: Select ‘Create a New Virtual Application Package.’
   2. At the Packaging Method screen, select ‘Create Package.’
   3. At the Type of Application Screen select ‘Add-on or Plug-in.’
   4. At Select Installer either select the installer file for the Add-on or Plug-in or select ‘Perform a custom installation’ just as in Section 8.
   5. Click the **Expand Package** button and point to the .appv file of the main application package to expand the package to the local system.
   6. At the **Install the primary parent program** screen, select the checkbox “I have installed the primary parent program” (this was already done in Step 3).
   7. At the **Package Name** screen, create a unique name for the add-on or plug-in package.
   8. At the **Installation** screen, install the plug-in. When the installation is complete, return to the sequencer and select the ‘I am finished installing’ check box and click ‘Next.’
   9. Complete the package.
4. Create a Connection Group on the App-V Management Server to link the two packages
   1. In the left-hand tree of the **App-V Server Console** select **Packages > Connection Groups**.
   2. Click the **ADD CONNECTION GROUP** link in the upper-right corner.

**Note:** **New Connection Group** appears in the list.

* 1. Click on the name of the **New Connection Group,** a rename box appears. Rename the connection group with a name that helps to properly identify it.
  2. In the lower pane, next to the **CONNECTED PACKAGES** header, click the **EDIT** link.
  3. In the lower right **PACKAGES** pane:
     1. Highlight the first package to add to the group and click the **Left Arrow** button.
     2. Highlight the next package to add to the group and click the **Left Arrow** button.
     3. Repeat for any additional package to add to the group.
     4. Click **Apply.**
     5. Click **Close.**
  4. In the lower pane, next to the **AD ACCESS** header, click the **EDIT** link.
  5. In the text field in the lower pane, enter **Domain\Group** to specify which AD groups should receive this connection group and click **Grant Access.**
  6. Click **Close.**
  7. In the upper pane, right-click the Connection Group and select **Publish.**

## Middleware

When creating a separate package to contain middleware to be available to other primary applications, for example, JavaTM or the Oracle® client, select the ‘Middleware’ option.

The work flow for middleware is:

1. **Sequence** the middleware application (Type of Application: Middleware) and copy package files to storage
   1. Follow the same process in [Section 7](#_Sequencing_Walk_Through) of this document.
2. **Revert** the sequencing virtual machine back to the base snapshot.
3. Sequence the Middleware App
   1. In the Sequencer: Select **Create a New Virtual Application Package**.
   2. At the **Packaging Method** screen, select **Create Package**.
   3. At the **Type of Application Screen** select: **Middleware**.
   4. At **Select Installer** either select the installer file for the **Primary Application** or select **Perform a custom installation** just as in [Section 7](#_Sequencing_Walk_Through).
   5. At the **Package Name** screen, create a unique name for the Primary Application.
   6. At the **Installation** screen, install the **Middleware** app. When installation is complete, back in the sequencer select the **I am finished installing** check box and click **Next**.
   7. Complete the package.
4. Create a Connection Group on the App-V Management Server to link the two packages
   1. In the left-hand tree of the **App-V Server Console** select **Packages > Connection Groups**.
   2. Click the **ADD CONNECTION GROUP** link in the upper-right corner.

**Note:** **New Connection Group** appears in the list.

* 1. Click on the name of the **New Connection Group,** a rename box appears. Rename the connection group with a name that helps to properly identify it.
  2. In the lower pane, next to the **CONNECTED PACKAGES** header, click the **EDIT** link.
  3. In the lower right **PACKAGES** pane:
     1. Highlight the first package to add to the group and click the **Left Arrow** button.
     2. Highlight the next package to add to the group and click the **Left Arrow** button.
     3. Repeat for any additional package to add to the group.
     4. Click **Apply.**
     5. Click **Close.**
  4. In the lower pane, next to the **AD ACCESS** header, click the **EDIT** link.
  5. In the text field in the lower pane, enter **Domain\Group** to specify which AD groups should receive this connection group and click **Grant Access.**
  6. Click **Close.**
  7. In the upper pane, right-click the Connection Group and select **Publish.**

# Sequencer Templates

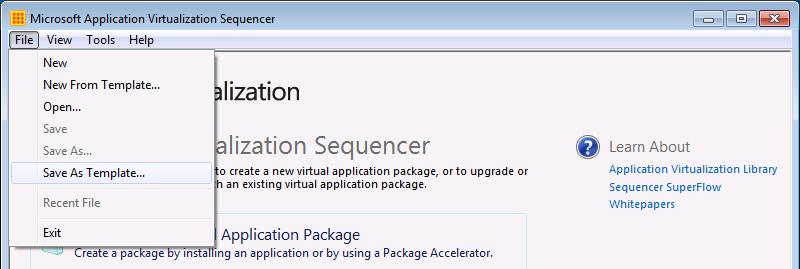
The App-V 5 Sequencer provides the ability to create sequencer template files that help automate and standardize specific settings for the sequencing process. In environments where the Sequencer’s default configuration is not best suited, Sequencer templates allow the engineer to save frequently changed sequencing options in the areas listed in Figure 29. The engineer can simply apply the Sequencer template before beginning sequencing in order to save time.

Create a template for changes specific to configuration options within the **Tools | Options** menu (ex: General sequencer settings, Parse items and Exclusion items) the engineer can make those changes at any time and select **File | Save As Template** to save the template file.

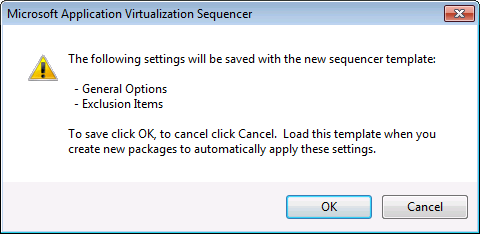
Templates are also very important for upgrade scenarios. The Sequencer does not save state so when a new Sequencer session is open and a package is opened for upgrade, the settings are in the default state. If certain sequencer settings were changed when sequencing a package, the changes will not remain at time of upgrade. Therefore, it is recommended to save a template for any package that has Sequencer customizations, and re-apply them on upgrade. A template may also contain additional options such as Package Deployment Settings and Advanced Monitoring Options.

To create a template select **File | Save As Template** upon completion of a particular sequencing.

To Apply a template select **File | New From Template** immediately after opening the Sequencer.



**Figure 36: Save as Template**



**Figure 37: Template Information**

1. **App-V Package Converter**

Virtual Application Packages created using any sequencer prior to version 5 cannot be published on the App-V 5 server, or be received by App-V 5 Clients. The App-V 5 Sequencer provides the App-V Package Converter as a tool to convert legacy virtual applications packages into the App-V v5 format.

The Package Converter can only convert a package sequenced with version 4.5 or 4.6 of the App-V Sequencer. Any packages sequenced with any version of the App-V Sequencer prior to 4.5 must first be opened and re-saved using the App-V 4.5 or 4.6 sequencer before it can be processed by the App-V 5 Package Converter.

The tool is entirely based in Windows PowerShell and functions completely independent of the App-V Sequencer application, therefore the Package Converter commands can be used manually or scripted in order to automate the package conversion process.

**Note:** In the 64-bit version of the Sequencer, the Package Converter cmdlets exist in the PowerShell (x86) shell.

## Package Converter PowerShell commands

***Command:* Test-AppvLegacyPackage**

This cmdlet takes as an input 4.5/4.6 packages, and performs basic validation of the package. The cmdlet returns a report containing any common errors found prior to conversion, so an administrator can discard or fix the package prior to conversion. Error warnings include but are not limited to:

* Important package files missing (.OSD, .SFT, etc.)
* OSD files containing scripts. (Scripts cannot be converted by the Package Converter)
* Package version not convertible (4.2 or lower)

**Note:** A successful report does not necessarily guarantee a successful conversion.

Syntax:  **Test-AppvLegacyPackage –SourcePath <path>**

Example: **Test-AppvLegacyPackage –SourcePath “\\AppV4Server\Content\LegacyPackage”**

If the testing shows no errors, the next step is to perform the conversion.

***Command:* ConvertFrom-AppvLegacyPackage**

This cmdlet takes as an input a 4.5/4.6 package, the target output directory, and converts the package into the new .APPV format.

Syntax: **ConvertFrom-AppvLegacyPacakge –SourcePath <path> -DestinationPath <path>**

Example: **ConvertFrom-AppvLegacyPackage –SourcePath “\\AppV4Server\Content\LegacyPackage” -DestinationPath “\\AppV5Server\Content\NewPackage”**

**Note:** The Destination Path must exist before running the cmdlet.

Since the Package Converter is based in PowerShell, it is possible to use any number of scripting methods to automate the testing and conversion of multiple legacy packages. Below is just one example:

* **Get-ChildItem “\\Appv4Server\Content” | ConvertFrom-AppvLegacyPackage -DestinationPath “\\Appv5Server\Content”**

This command will parse all subfolders within the legacy content share and convert all packages therein placing them all in the Destination Path.

# Dynamic Configuration and Targeted Scripting

App-V 5.0 introduces a new management process for packages called Dynamic Configuration. Dynamic Configuration allows for specifying a policy for a package at either the machine level or at the user level. The Dynamic Configuration files enable sequencing engineers 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 make modifications to the application to provide proper functionality within the App-V environment. One of the most common methods is making modifications to the \_\****configuration.xml*** files to allow certain actions to be performed at a specified time during the execution of the application. For example it may be desired to disable a mailto extension to prevent a virtualized application from overwriting that extension from another application. Or perhaps there is an application that requires access to a specific mapped drive and that mapped drive must follow the user.

Scripting in these Dynamic Configuration files is an invaluable tool when sequencing / publishing applications. The scripts are xml based and can be used to setup or alter the virtual environment as well as execute scripts at time of deployment or removal, before an application executes, or they can be used to “clean up” the environment after the application terminates. These custom modifications can be made using any text editor or XML editor utility.

In addition, Dynamic Configuration allows flexibility to control specific configuration changes or scripts to apply in a machine context (global—across all users who access the app on the machine), or at a user level (users in specific AD groups can have different custom configurations for the same application on a machine). Dynamic Configuration files allow running scripts at various execution times of the package lifecycle. In section 14.1.2 is a table that helps describe the various script events and when and how they can execute.

## Deployment Configuration Files

The deployment configuration file and user configuration file are generated automatically by the sequencer and are called ***PackageName\_DeploymentConfig.xml*** and ***PackageName\_UserConfig.xml*** respectively.

* **DeploymentConfig** files are applied to the package, which can be deployed to AD groups containing Machine or User accounts.
* **UserConfig** files are applied to the package to AD groups containing User accounts.

### DeploymentConfig vs. UserConfig.

Before leveraging DeploymentConfig and UserConfig files an understanding of the capabilities is required. This includes when customization should occur, what will be customized, and where the customization is to be applied.

**UserConfig** files provide configuration settings that can be applied 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)
* Managing Authority (for controlling co-existence of package with App-V 4.6)

**UserConfig** files are published to AD groups containing **users** and SCCM collections based on **users.** Each package can have multiple associated UserConfig files, where a file is published to specific user groups to customize the configuration of the application. The registry settings are applied to **HKCU** and file opeations (e.g. shortcuts) are saved in per-user locations.

**DeploymentConfig** files provide 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 UserConfig list above:

* All UserConfig settings above
* Extensions that can only be applied globally for all users
* Virtual Subsystems that can be configured for global machine locations e.g. registry
* Product Source URL
* Scripts (Machine context only)
* Controls to Terminate Child Processes

One **DeploymentConfig** file is assigned to a package and is delivered to the user or computer based on the assigned group scope (user or computer). These two different target options store settings in different locations:

* Packages published to an AD Group containing machine accounts (global) store registry settings in **HKLM** and created files (e.g. shortcuts) in global locations.
* Packages published to an AD group containing user accounts store machine context settings in the registry to **HKLM** and files operations are saved to global locations. Extensions under the machine context are only applied if the package is deployed globally to a machine account. Modifications defined in the user context are applied to **HKCU** and file operatons (e.g. shortcuts) are saved in per-user locations**.**

### Comparing Script processing in Deployment and User Config files

Deploying scripts with Dynamic Configuration files provides options for the timing of the execution of the script called. DeploymentConfig files are a superset of the userconfig files. The deploymentconfig has additional triggers and user context in addition to machine context. Based on when the desired script needs to be called and/or what context the script must run under will determine the dynamic configuration file (DeploymentConfiguration, UserConfiguration) where the script is implemented. The table below describes the various script events and the context under which they can run.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Script Execution Time | Can be specified in Deployment Configuration  (Script runs as) | Can be specified in User Configuration  (Script runs as) | Can run in the Virtual Environment of the package | Can be run in the context of a specific application |
| AddPackage | X (System Account) |  |  |  |
| PublishPackage | X (System Account) | X (Current User) |  |  |
| UnpublishPackage | X (System Account) | X (Current User) |  |  |
| RemovePackage | X (System Account) |  |  |  |
| StartProcess | X (Current User) | X (Current User) | X | X |
| ExitProcess | X (Current User) | X (Current User) |  | X |
| StartVirtualEnvironment | X (Current User) | X (Current User) | X |  |
| TerminateVirtualEnvironment | X (Current User) | X (Current User) |  |  |

Dynamic Configuration enables packages with drivers that require extraction and deployment in order function correctly. Since App-V does not virtualize drivers, the driver must be installed natively. Rather than pushing out to all users or attempting to write the installation in a separate process, a script can be called from the **Deployment** Configuration file at the “PublishPackage” or “AddPackage” event in the **Deployment Configuration** file. Calling the script from either event will ensure that the driver has been made available to the user by the time the package is. Here is an example:

<PublishPackage>

<Path>powershell.exe </Path>

<Arguments>-file \\server\scripts\installDriver.ps1</Arguments>

</PublishPackage>

**Note: Review section 14.3.1 for additional detail on referencing scripts from specific locations.**

It is important to remember that if this script was deployed via a **User Configuration** file, that script is functioning under the context of the current **user.** As a result, if what the script is trying to perform requires elevated privileges such as in the driver installation example, the script may not succeed or the user may be prompted to enter elevated credentials if the user does not already possess them. In this example, deploying the driver installation script via a **Deployment Configuration** file, published to the machine rather than the user and ensured that the script functioned under the system context and possessed the necessary permission to deploy the hardware driver.

## Enabling Scripting

Before implementing dynamic configuration scripting, the App-V Client must be enabled to process scripts. This can be done by setting the client configuration setting “EnablePackageScripts”. This can be done at install time, through group policy or through PowerShell. Below is the command for enabling scripting from PowerShell:

**PS > Set-AppvClientConfiguration -EnablePackageScripts 1**

## Writing a Script

This section provides further detail on how to add scripting to Dynamic Configuration files.

### Referencing script collateral

The path element represents the executable (**the binary specified must be an executable file**) that will be launched when the script is run. The path can be a local or UNC path or can be a file embedded in the package. The full path must be specified when accessing local system files or UNC files. For accessing files inside of the package, specify a **relative path** or a **tokenized path**.

The working directory of a script is the base directory of the package (e.g. C:\ProgramData\App-V\A2D337CF-BAFD-4A51-A385-466B9E6053A7\B83CE87D-558A-4C20-BF23-B663BF19D922) which enables specifying a script file with a relative path inside the package by doing the following:

<PublishPackage>

<Path>powershell.exe </Path>

<Arguments>**.\Scripts**\InstallDriver.ps1 </Arguments>

<Wait RollbackOnError="true" Timeout="120"/>

</PublishPackage>

Or, since the Script node and the Package root are siblings, use a relative path from the [{AppVPackageRoot}] token:

<PublishPackage>

<Path>powershell.exe </Path>

<Arguments**>[{AppVPackageRoot}]\..\Scripts**\InstallDriver.ps1 </Arguments>

<Wait RollbackOnError="true" Timeout="120"/>

</PublishPackage>

Additional information on the tokens available in App-V 5.0 is later in this document in the section titled [**App-V Sequencer Tokens**](#_App-V_Sequencer_Tokens).

### Blocking Logic

The <**Wait**> element can be used to introduce synchronization logic if the script is critical or necessary in order for the package to work correctly. For example, in the driver scenario above, the application may not function correctly if the driver is not properly installed. “**RollbackOnError**” should be set to “**true**” in this case since the package is nonfunctional without successful script completion. This introduces additional logic that may lock the publishing process. App-V allows a timeout in **seconds** indicating how long the event should wait for the process to exit. In this case, it is set to two minutes for the installer to run, if it exits sooner, the event (publish in this case) will complete. If it does not exit in time, it will fail.

In other scenarios where the script is non-critical, the script does cleanup of some user data files if they are present, but it is not critical for the package functionality. This example presents a reason to set “**RollbackOnError”** to “**false**” or remove the <Wait> element altogether.

**Note:** The “**Timeout”** default is ‘0’ and does not need to be specified. A timeout of ‘0’ means “wait indefinitely” for the process to exit.

## Usage Scenario

An organization has virtualized a core line of business application. After deploying the application it is discovered that the application is overriding the “mailto” URL Protocol handler from locally installed Microsoft Office and is not the desired behavior. Since this configuration change needs to apply to all users on the targeted machines, the DeploymentConfig file is modified and published as the default configuration.

The required configuration change in the **DeploymentConfig.xml** file::

* Search for **<URLProtocols Enabled=”true”/>** and changeto **<URLProtocols Enabled=”false”/>**

After this issue has been resolved, another specific group of users identifies a mapped drive as a requirement. Rather than creating a persistent mapping of that drive on the system, scripts called dynamic configuration files are utilized to map during startup and disconnect the drive at shutdown of the application. Because this configuration is specific to only one group of users, a **UserConfig.xml** file is modified and published to the specific group of users that require the drive mapping.

Create two simple BAT files that **“Map”** and **“Disconnect”** the drive and place the BAT files on the [**\\appvserver\content**](file:///\\appvserver\content) share. Add the scripts to the **UserConfig.xml** file performing the following modifications:

* Between the **</Applications>** and **</UserConfiguration>** tags, insert the following:

<UserScripts>

<StartVirtualEnvironment RunInVirtualEnviornment=”false”>

<Path>**%SYSTEMROOT%\System32\CMD.EXE**</Path>

<Arguments>**/K “\\appvserver\content\Map.bat”**<**/**Arguments>

</StartVirtualEnvironment>

<TerminateVirtualEnvironment RunInVirtualEnvironment=”false”>

<Path>**%SYSTEMROOT%\System\CMD.EXE**</Path>

<Arguments>**/K “\\appvserver\content\Disconnect.bat”**</Arguments>

</TerminateVirtualEnvironment>

</UserScripts>

## Script Examples.

This section provides examples of the xml syntax for each of the event timings when referencing scripts in Dynamic Configuration files. This is a reference to all of the timing events in Dynamic Configuration files, for example purposes.

Adding scripts to virtual application packages that are published relative to specific user groups, the modifications need to be made to the <UserScripts> section of the **packagename\_UserConfig.xml** file:

<UserScripts>

<StartProcess RunInVirtualEnvironment="true">

<Path>VFS\ProgramFilesX86\App\dosomething.exe</Path>

<Arguments>-WithArgs</Arguments>

<Wait RollbackOnError="true"/>

<ApplicationId>{a56fa627-c35f-4a01-9e79-7d36aed8225a}</ApplicationId>

</StartProcess>

<ExitProcess>

<Path>VFS\ProgramFilesX86\App\UnDoSomething.exe</Path>

<Arguments>-WithArgs</Arguments>

<Wait RollbackOnError="false"/>

<ApplicationId>{a56fa627-c35f-4a01-9e79-7d36aed8225a}</ApplicationId>

</ExitProcess>

<StartVirtualEnvironment RunInVirtualEnvironment="true">

<Path>[{PackageRoot}]\VFS\ProgramFilesX86\App\DoSomething.exe</Path>

<Arguments>-WithArgs</Arguments>

<Wait RollbackOnError="true"/>

</StartVirtualEnvironment>

<TerminateVirtualEnvironment>

<Path>[{PackageRoot}]\VFS\ProgramFilesX86\App\UnDoSomething.exe</Path>

<Arguments>-WithArgs</Arguments>

<Wait RollbackOnError="false"/>

</TerminateVirtualEnvironment>

<PublishPackage>

<Path>\\server\share\DoSomething.exe</Path>

<Arguments>-WithArgs</Arguments>

<Wait RollbackOnError="true" Timeout="30"/>

</PublishPackage>

<UnpublishPackage>

<Path>\\server\share\UnDoSomething.exe</Path>

<Arguments>-WithArgs</Arguments>

<Wait RollbackOnError="false" Timeout="30"/>

</UnpublishPackage>

</UserScripts>

Adding scripts to virtual application packages that are published relative to specific machines (global publishing), the modifications need to be made to the <MachineScripts> section of the **packagename\_DeploymentConfig.xml** file:

<MachineScripts>

<PublishPackage>

<Path>\\server\share\DoSomething.exe</Path>

<Arguments>-WithArgs</Arguments>

<Wait RollbackOnError="true" Timeout="30"/>

</PublishPackage>

<UnpublishPackage>

<Path>\\server\share\UnDoSomething.exe</Path>

<Arguments>-WithArgs</Arguments>

<Wait RollbackOnError="false" Timeout="30"/>

</UnpublishPackage>

<AddPackage>

<Path>[{PackageRoot}]\VFS\ProgramFilesX86\App\InstallDriver.exe</Path>

<Arguments>-DriverName Printer-Visibility Northamerica</Arguments>

<Wait RollbackOnError="true" Timeout="30"/>

</AddPackage>

<RemovePackage>

<Path>[{PackageRoot}]\VFS\ProgramFilesX86\App\InstallDriver.exe</Path>

<Arguments>-Uninstall</Arguments>

<Wait RollbackOnError="false" Timeout="60"/>

</RemovePackage>

</MachineScripts>

## Complete settings available in the Dynamic Configuration files

This document will explain the sample dynamic configuration files generated by the sequencer. Additional information is available from the App-V documentation available at: <http://go.microsoft.com/fwlink/?linkid=269780> and additional guidance will be provided on dynamic configuration in the future.

### Sample *XmlNotepad\_*DeploymentConfig.xml

This is a sample dynamic deployment configuration created automatically by the sequencer for the XMLNotepad application. This file can be modified to override machine wide and per user elements within the sequenced package. This XML document is pre-filled with data from a sample sequenced package.

This is a reference for customizing the document with changes that are published along with the package.

There are two sections, a per-user Configuration (settings apply in the user context) and a per-machine configuration which applies in the machine context.

-->

<DeploymentConfiguration PackageId="dc19433c-2192-49c9-81f0-bc18ea321aa9" DisplayName="XMLNotepad" IgnorableNamespaces="" xmlns="http://schemas.microsoft.com/appv/2010/deploymentconfiguration">

<!--

Per User Configuration

-->

<UserConfiguration>

<!--

Managing Authority - Allow App-V to take over App-V 4.6 extension points for the named package.

-->

<ManagingAuthority TakeoverExtensionPointsFrom46="true" PackageName="dc19433c-2192-49c9-81f0-bc18ea321aa9" />

<Subsystems>

<!-- Integration subsystems are enabled by default -->

<!--OVERRIDE BEHAVIOR OF MANIFEST AND CONFIGURATION FILES

Integration Subsystems can be enabled and disabled independently of the content.

Thus if Shortcuts are enabled, the client will use the shortcuts contained within

the manifest by default.

Each Integration 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.

e.g. for the shortcuts subsystem,

(A)

if the user defined this in either the dynamic or deployment config file:

<Shortcuts Enabled="true">

<Extensions>

...

</Extensions>

</Shortcuts>

Content in the manifest will be ignored.

(B)

IF the user defined only this:

<Shortcuts Enabled="true"/>

Then the content in the Manifest will be integrated during publishing.

(C)

note that if the user defines this:

<Shortcuts Enabled="true">

<Extensions/>

</Shortcuts>

Then all the Shortcuts within the manifest will still be ignored.

There will be no shortcuts integrated.-->

<!--

Shortcuts

-->

<Shortcuts Enabled="true">

<Extensions>

<Extension Category="AppV.Shortcut">

<Shortcut>

<File>[{Desktop}]\XML Notepad 2007.lnk</File>

<Target>[{AppVPackageRoot}]\XmlNotepad.exe</Target>

<Arguments />

<Icon>[{AppVPackageRoot}]\ApplicationIcons\XML Notepad 2007 2.5.0.0.ico</Icon>

<WorkingDirectory />

</Shortcut>

</Extension>

<Extension Category="AppV.Shortcut">

<Shortcut>

<File>[{Programs}]\XML Notepad 2007\XML Notepad 2007.lnk</File>

<Target>[{AppVPackageRoot}]\XmlNotepad.exe</Target>

<Arguments />

<Icon>[{AppVPackageRoot}]\ApplicationIcons\XML Notepad 2007 2.5.0.0.ico</Icon>

<WorkingDirectory />

</Shortcut>

</Extension>

</Extensions>

</Shortcuts>

<!--

Virtual Registry

-->

<Registry Enabled="true">

<!--

<Include>

<Key Path="\REGISTRY\USER\Software\Foo">

<Value Type="REG\_SZ" Name="Bar" Data="NewValue"/>

</Key>

<Key Path="\REGISTRY\USER\Software\EmptyKey"/>

</Include>

-->

<!--

<Delete>

<Key Path="\REGISTRY\USER\Software\Foo\BarDelete"/>

</Delete>

-->

</Registry>

<!--

Virtual File System

-->

<FileSystem Enabled="true" />

</Subsystems>

<!--

Applications

-->

<Applications>

<Application Id="[{AppVPackageRoot}]\XmlNotepad.exe" Enabled="true">

<VisualElements>

<Name>XML Notepad 2007 2.5.0.0</Name>

<Icon>[{AppVPackageRoot}]\ApplicationIcons\XML Notepad 2007 2.5.0.0.ico</Icon>

<Description />

</VisualElements>

</Application>

</Applications>

<!-- User Scripts Example - customize and uncomment to use user scripts -->

<!--

<UserScripts>

<StartProcess RunInVirtualEnvironment="true">

<Path>VFS\ProgramFilesX86\App\dosomething.exe</Path>

<Arguments>-WithArgs</Arguments>

<Wait RollbackOnError="true"/>

<ApplicationId>{a56fa627-c35f-4a01-9e79-7d36aed8225a}</ApplicationId>

</StartProcess>

<ExitProcess>

<Path>VFS\ProgramFilesX86\App\UnDoSomething.exe</Path>

<Arguments>-WithArgs</Arguments>

<Wait RollbackOnError="false"/>

<ApplicationId>{a56fa627-c35f-4a01-9e79-7d36aed8225a}</ApplicationId>

</ExitProcess>

<StartVirtualEnvironment RunInVirtualEnvironment="true">

<Path>[{PackageRoot}]\VFS\ProgramFilesX86\App\DoSomething.exe</Path>

<Arguments>-WithArgs</Arguments>

<Wait RollbackOnError="true"/>

</StartVirtualEnvironment>

<TerminateVirtualEnvironment>

<Path>[{PackageRoot}]\VFS\ProgramFilesX86\App\UnDoSomething.exe</Path>

<Arguments>-WithArgs</Arguments>

<Wait RollbackOnError="false"/>

</TerminateVirtualEnvironment>

<PublishPackage>

<Path>\\server\share\foobar.exe</Path>

<Arguments>-WithArgs</Arguments>

<Wait RollbackOnError="true" Timeout="30"/>

</PublishPackage>

<UnpublishPackage>

<Path>\\server\share\barfoo.exe</Path>

<Arguments>-WithArgs</Arguments>

<Wait RollbackOnError="false" Timeout="30"/>

</UnpublishPackage>

</UserScripts>

-->

</UserConfiguration>

<!--

Machine Wide Configuration

-->

<MachineConfiguration>

<!--

Product Source URL Opt Out

-->

<ProductSourceURLOptOut Enabled="true" />

<Subsystems>

<!--

Machine Wide Virtual Registry

-->

<Registry>

<!--

<Include>

<Key Path="\REGISTRY\Machine\Software\Foo">

<Value Type="REG\_SZ" Name="Bar" Data="Baz"/>

</Key>

<Key Path="\REGISTRY\Machine\Software\EmptyKey"/>

</Include>

-->

<!--

<Delete>

<Key Path="\REGISTRY\Machine\Software\Foo\BarDelete"/>

</Delete>

-->

</Registry>

</Subsystems>

<!-- Machine Scripts Example - customize and uncomment to use machine scripts -->

<!--

<MachineScripts>

<PublishPackage>

<Path>\\server\share\DoSomething.exe</Path>

<Arguments>-WithArgs</Arguments>

<Wait RollbackOnError="true" Timeout="30"/>

</PublishPackage>

<UnpublishPackage>

<Path>\\server\share\UnDoSomething.exe</Path>

<Arguments>-WithArgs</Arguments>

<Wait RollbackOnError="false" Timeout="30"/>

</UnpublishPackage>

<AddPackage>

<Path>[{PackageRoot}]\VFS\ProgramFilesX86\App\InstallDriver.exe</Path>

<Arguments>-DriverName Printer-Visibility Northamerica</Arguments>

<Wait RollbackOnError="true" Timeout="30"/>

</AddPackage>

<RemovePackage>

<Path>[{PackageRoot}]\VFS\ProgramFilesX86\App\InstallDriver.exe</Path>

<Arguments>-Uninstall</Arguments>

<Wait RollbackOnError="false" Timeout="60"/>

</RemovePackage>

</MachineScripts>

-->

</MachineConfiguration>

</DeploymentConfiguration>

# Advanced Sequencing Techniques

This section will briefly cover some sequencing techniques and troubleshooting information. Learning the logging options during sequencing, and the specific functions and information presented is recommended.

## Sequencing Web-Based Applications

Sequencing web based applications, such as an Active X control is very similar to sequencing a traditional application like Microsoft Office. In these examples, one of the limitations of App-V is important: Internet Explorer is NOT supported in a sequenced application. However, Plug-ins or ActiveX controls for Internet Explorer can be virtualized.

**Figure 16: Sequencing Web-Based Applications Process**

Following the process listed in the figure provides a package that can be deployed to client computers and utilizes the locally installed Internet Explorer to support the virtual package. This pulls the locally installed application into the virtual environment with the specified additions, in this case, a Web plug-in. This enables a clean and secure Internet Explorer that is locked down on a user’s machine and allows plug-ins that are deployed as services and services separately.

1. **Additional Information**

## Built-in reporting, Error Codes, and the Microsoft Knowledge Base

Troubleshoot sequencing with the built-in reporting, error codes, and the Microsoft support site. Additional troubleshooting can be completed by reaching out to software vendors and independent sequencing and packaging sites.

**Review the report.xml after Sequencing to identify detected issues.**

The Report.xml file contains valuable information, warnings, and errors that occur during the Sequencing process. Although it is not exhaustive (no errors in the file does not mean it is successful), use information in the report to assist in refining or fixing a package.

The virtual application should be tested on a client machine until the application functionality is verified.

**Sequencer writes messages to the Event Viewer.**

Go to Start, type Event Viewer, and open the Event Viewer. Select the Applications and Services Log, select Microsoft, select AppV, select Sequencer. Utilize the messages in the event viewer to diagnose a Sequencer failure issue.

**The Microsoft Knowledge Base.**

The Microsoft knowledge base contains useful information regarding planning, implementing, and troubleshooting App-V. A link to the Microsoft knowledge base follows:

<http://support.microsoft.com>

*Error Code Formats*: When searching the knowledge base for error codes, be aware that the first 6 digits are unique to the version of the App-V Client software. The ending 10 digits of the error code contain information pertinent to the error.

When searching for an error code, enter the last 8 digits such as 00002AF9.

e.g., xxxxxx - xxxxxxxxxx

or

The first block of 6 characters identify

* File ID
* Line

In the Next Block of 10 Characters

* The first 2 digits indicate the module
* The last 8 digits indicate the error

## App-V Sequencer Tokens

App-V Sequencer comes with many tokens. These tokens can be viewed in the Tools 🡪 Options dialog box, under Parse Items. A list of tokens is provided below:

Each of these tokens are parenthesized with a [{ and }].

|  |  |
| --- | --- |
| Known Folder Token | Known Folder Path |
| AccountPictures | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\AccountPictures |
| Administrative Tools | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Administrative Tools |
| AppData | C:\Users\<username>\AppData\Roaming |
| Application Shortcuts | C:\Users\<username>\AppData\Local\Microsoft\Windows\Application Shortcuts |
| Cache | C:\Users\<username>\AppData\Local\Microsoft\Windows\Temporary Internet Files |
| CD Burning | C:\Users\<username>\AppData\Local\Microsoft\Windows\Burn\Burn |
| Common Administrative Tools | C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Administrative Tools |
| Common AppData | C:\ProgramData |
| Common Desktop | C:\Users\Public\Desktop |
| Common Documents | C:\Users\Public\Documents |
| Common Programs | C:\ProgramData\Microsoft\Windows\Start Menu\Programs |
| Common Start Menu | C:\ProgramData\Microsoft\Windows\Start Menu |
| Common Startup | C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup |
| Common Templates | C:\ProgramData\Microsoft\Windows\Templates |
| CommonDownloads | C:\Users\Public\Downloads |
| CommonMusic | C:\Users\Public\Music |
| CommonPictures | C:\Users\Public\Pictures |
| CommonRingtones | C:\ProgramData\Microsoft\Windows\Ringtones |
| CommonVideo | C:\Users\Public\Videos |
| Contacts | C:\Users\<username>\Contacts |
| Cookies | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Cookies |
| CredentialManager | C:\Users\<username>\AppData\Roaming\Microsoft\Credentials |
| CryptoKeys | C:\Users\<username>\AppData\Roaming\Microsoft\Crypto |
| Desktop | C:\Users\<username>\Desktop |
| Device Metadata Store | C:\ProgramData\Microsoft\Windows\DeviceMetadataStore |
| DocumentsLibrary | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Libraries\Documents.library-ms |
| Downloads | C:\Users\<username>\Downloads |
| DpapiKeys | C:\Users\<username>\AppData\Roaming\Microsoft\Protect |
| Favorites | C:\Users\<username>\Favorites |
| Fonts | C:\windows\Fonts |
| GameTasks | C:\Users\<username>\AppData\Local\Microsoft\Windows\GameExplorer |
| History | C:\Users\<username>\AppData\Local\Microsoft\Windows\History |
| ImplicitAppShortcuts | C:\Users\<username>\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch\User Pinned\ImplicitAppShortcuts |
| Libraries | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Libraries |
| Links | C:\Users\<username>\Links |
| Local AppData | C:\Users\<username>\AppData\Local |
| LocalAppDataLow | C:\Users\<username>\AppData\LocalLow |
| MusicLibrary | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Libraries\Music.library-ms |
| My Music | C:\Users\<username>\Music |
| My Pictures | C:\Users\<username>\Pictures |
| My Video | C:\Users\<username>\Videos |
| NetHood | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Network Shortcuts |
| Personal | C:\Users\<username>\Documents |
| PicturesLibrary | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Libraries\Pictures.library-ms |
| Podcast Library | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Libraries\Podcasts.library-ms |
| Podcasts | C:\Users\<username>\Podcasts |
| PrintHood | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Printer Shortcuts |
| Profile | C:\Users\<username> |
| ProgramFiles | C:\Program Files |
| ProgramFilesCommon | C:\Program Files\Common Files |
| ProgramFilesCommonX64 | C:\Program Files\Common Files |
| ProgramFilesCommonX86 | C:\Program Files (x86)\Common Files |
| ProgramFilesX64 | C:\Program Files |
| ProgramFilesX86 | C:\Program Files (x86) |
| Programs | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs |
| Public | C:\Users\Public |
| PublicAccountPictures | C:\Users\Public\AccountPictures |
| PublicGameTasks | C:\ProgramData\Microsoft\Windows\GameExplorer |
| PublicLibraries | C:\Users\Public\Libraries |
| Quick Launch | C:\Users\<username>\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch |
| Recent | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Recent |
| RecordedTVLibrary | C:\Users\Public\Libraries\RecordedTV.library-ms |
| ResourceDir | C:\windows\resources |
| Ringtones | C:\Users\<username>\AppData\Local\Microsoft\Windows\Ringtones |
| Roamed Tile Images | C:\Users\<username>\AppData\Local\Microsoft\Windows\RoamedTileImages |
| Roaming Tiles | C:\Users\<username>\AppData\Local\Microsoft\Windows\RoamingTiles |
| SavedGames | C:\Users\<username>\Saved Games |
| Searches | C:\Users\<username>\Searches |
| SendTo | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\SendTo |
| Start Menu | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Start Menu |
| Startup | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup |
| System | C:\windows\system32 |
| SystemCertificates | C:\Users\<username>\AppData\Roaming\Microsoft\SystemCertificates |
| SystemX86 | C:\windows\SysWOW64 |
| Templates | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Templates |
| User Pinned | C:\Users\<username>\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch\User Pinned |
| UserProfiles | C:\Users |
| VideosLibrary | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Libraries\Videos.library-ms |
| Windows | C:\windows |
| Custom Token | Custom Token Expansion |
| AppVAllUsersDir | C:\Users\All Users |
| AppVComputerName | <USERNAME>-LT02 |
| AppVCurrentUserSID | S-1-5-21-124525095-708259637-1543119021-705252 |
| AppVEnvironmentVariableCommonProgramFiles | %commonprogramfiles% |
| AppVEnvironmentVariableProgramFiles | %ProgramFiles% |
| AppVPackageDrive | C: |
| AppVPackageRoot | C:\AppInstallFolder |
| AppVSystem32Catroot | C:\windows\system32\catroot |
| AppVSystem32Catroot2 | C:\windows\system32\catroot2 |
| AppVSystem32DriversEtc | C:\windows\system32\drivers\etc |
| AppVSystem32Driverstore | C:\windows\system32\driverstore |
| AppVSystem32Logfiles | C:\windows\system32\logfiles |
| AppVSystem32Spool | C:\windows\system32\spool |

## App-V Team Blog

The App-V Team Blog

is probably the single greatest repository of App-V related information on the web. This should be the first stop for anyone interested in learning about App-V and sequencing. The blog is updated frequently so keep checking it for more information. Additionally, look through the archives for valuable information.

* App-V Team Blog - <http://blogs.technet.com/b/appv/>
* RSS Feed - [http://blogs.technet.com/App-V/rss.xml](http://blogs.technet.com/softgrid/rss.xml)

## App-V Technical Discussion Forums and Web Sites

Here is a list of non-Microsoft sites that contain information related to the App-V product and sequencing. Some of these sites include useful App-V related technical discussion forums. While these sites contain valuable information about App-V, it should be noted that these sites are not affiliated with Microsoft in any way and Microsoft cannot verify any of the information contained within.

* http://www.appvirtguru.com
* <http://www.stealthpuppy.com>
* <http://www.itninja.com> (formerly AppDeploy.com)
* <http://blogs.technet.com/virtualworld>
* <http://social.technet.microsoft.com/Forums/en-US/category/appvirtualization>