

**Microsoft Application Virtualization 4.6 SP1**

**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, an example of sequencing, important information related to updating packages, and finally, examples of advanced OSD scripting.

Copyright © 2011 MICROSOFT CORPORATION

Contents

[1 Introduction to Application Sequencing 5](#_Toc286934407)

[2 Sequencer Workstation Configuration 6](#_Toc286934408)

[3 New features of App-V 4.6 SP1 Sequencer 8](#_Toc286934409)

[3.1 Changes to the Sequencer in 4.6 SP1 8](#_Toc286934410)

[3.1.1 Improved User Interface 8](#_Toc286934411)

[3.1.2 Package Accelerators 8](#_Toc286934412)

[3.1.3 Sequencer Templates 9](#_Toc286934413)

[3.1.4 Diagnostics (Reporting) 9](#_Toc286934414)

[3.1.5 Command Line Sequencer Enhancements 9](#_Toc286934415)

[4 Recommended Best Practices for Sequencing 10](#_Toc286934416)

[5 Classifying Applications for Sequencing 12](#_Toc286934417)

[6 Sequencing Limitations 13](#_Toc286934418)

[7 Sequence files 14](#_Toc286934419)

[8 Sample Sequencing 15](#_Toc286934420)

[8.1 Welcome Screen 15](#_Toc286934421)

[8.2 Packaging Method 16](#_Toc286934422)

[8.3 Prepare Computer 17](#_Toc286934423)

[8.4 Type of Application 18](#_Toc286934424)

[8.5 Select Installer 19](#_Toc286934425)

[8.6 Package Name 20](#_Toc286934426)

[8.7 Installation 21](#_Toc286934427)

[8.7.1 Applications that Require a Reboot 21](#_Toc286934428)

[8.8 Configure Software 22](#_Toc286934429)

[8.9 Installation Report 23](#_Toc286934430)

[8.10 Customize 23](#_Toc286934431)

[8.11 Edit Shortcuts 25](#_Toc286934432)

[8.12 Prepare for Streaming 27](#_Toc286934433)

[8.13 Target OS 28](#_Toc286934434)

[8.14 Create Package 30](#_Toc286934435)

[9 Creating a Package Accelerator 32](#_Toc286934436)

[9.1 Work-flow 32](#_Toc286934437)

[10 Create a package using a Package Accelerator 38](#_Toc286934438)

[10.1 Work-flow 38](#_Toc286934439)

[11 Application Package Upgrade 39](#_Toc286934440)

[11.1 Updating a Package that will Replace the Existing Package 40](#_Toc286934441)

[11.2 Updating a Package that will Run at the Same Time as the Existing Package 41](#_Toc286934442)

[12 Dynamic Suite Composition 42](#_Toc286934443)

[12.1 Advantages 42](#_Toc286934444)

[12.2 Workflow 43](#_Toc286934445)

[12.3 Add-on or Plug-in 44](#_Toc286934446)

[12.4 Middleware 45](#_Toc286934447)

[12.5 Dynamic Suite Composition Tool 45](#_Toc286934448)

[12.6 Manually Defining the Dependency in the Primary Package 49](#_Toc286934449)

[13 Sequencer Templates 49](#_Toc286934450)

[14 Advanced OSD Scripting 51](#_Toc286934451)

[14.1 OSD Scripts 52](#_Toc286934452)

[14.2 OSD: SCRIPTBODY vs. HREF 54](#_Toc286934453)

[14.3 <SCRIPTBODY> Examples 55](#_Toc286934454)

[14.4 <HREF> Examples 56](#_Toc286934455)

[14.5 <PARAMETERS> Examples 57](#_Toc286934456)

[14.6 <WORKINGDIR> Examples – Specifying a Working Directory 57](#_Toc286934457)

[14.7 <REGISTRY> Examples – Registry Customizations 58](#_Toc286934458)

[14.8 Launching Local Applications 58](#_Toc286934459)

[14.9 Setting Environment Variables 59](#_Toc286934460)

[14.10 Launching Application via Non-Default Provider Policy 59](#_Toc286934461)

[14.11 Running an Application as a CONSOLE application 60](#_Toc286934462)

[14.12 OSD XML Syntax Check 61](#_Toc286934463)

[15 Advanced Sequencing Techniques 62](#_Toc286934464)

[15.1 Sequencing Applications That Cannot Install to Q:\ 62](#_Toc286934465)

[15.2 Sequencing Web-Based Applications 63](#_Toc286934466)

[15.3 Sequencing Applications that Require Access to Local Devices and/or Resources 63](#_Toc286934467)

[15.4 Sequencer Log Files 64](#_Toc286934468)

[16 Finding Additional Information 65](#_Toc286934469)

[16.1 Error Codes and the Microsoft Knowledge Base 65](#_Toc286934470)

[16.2 App-V Team Blog 66](#_Toc286934471)

[16.3 App-V Related Technical Discussion Forums and Web Sites 66](#_Toc286934472)

# 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 key to a successful App-V implementation. As such, it’s important to follow Microsoft’s recommended practices and be aware of the different options when sequencing. The release of the App-V 4.6 SP1 Sequencer includes a number of powerful new features such as Package Accelerators, Settings Templates, and built-in diagnostics designed to make the process of sequencing faster and easier. This document covers setting up the Sequencer, sequencing best practices, an example of sequencing, updating, the new features, and finally, examples of advanced OSD 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:**

* **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 be required to be sequenced on the same OS environment that the App-V Client is running on in order to resolve those problems.
* **Include Microsoft Office on the base sequencer image if Microsoft Office will be installed traditionally on all desktops in the organization.** Many applications will install differently if they recognize that Microsoft Office is already installed on the machine. Thus, if an application is expected to integrate with Microsoft Office, it’s best to attempt sequencing on a machine with Office already installed and activated. This assumes that a Microsoft Office suite will be installed locally on all client PCs. In addition, you may want to install any other programs that could be used by the application you are sequencing if they are not going to be a part of the sequence.
* **Select the sequencing drive letter assignment.** The 4.6 SP1 Sequencer installer will mount a drive that will serve as the root for all packages. Specify the drive letter that matches the drive letter specified on the App-V client. You are no longer required to create a disk partition or separate physical disk as you did with previous versions.
* **Configure the temp directories to ensure successful sequencing and improve performance.** The Sequencer uses the %TMP%, %TEMP%, and its own Scratch Directory for temporary files. These locations should contain free disk space equivalent to the estimated installation size. The Scratch Directory is where the Sequencer will temporarily store files generated during the sequencing process. You can check the location of the Scratch Directory by launching the Sequencer, clicking Options from the Tools menu, and then noting the Scratch Directory box in the General tab. Placing the temp directories and the Scratch Directory on different hard drive spindles can improve performance during sequencing.
* **Sequence using Virtual PC.** Many applications will be sequenced more than once. This may be due to additional configuration changes or simply starting over to correct a mistake. The point is that you will be going back to your original configuration on the PC several times. To help facilitate this you may want to use a virtual machine. This will let you sequence an application and, with a simple click of a button, revert back to a clean state so you can continue sequencing with no down time. Additionally, whenever you start a new sequence, start with a clean sequencing system.
* **Shut down Other Programs.** Processes and scheduled tasks that normally run on your computer can slow down the sequencing process and cause irrelevant data to be gathered during sequencing. The Sequencer will scan for these and other processes before creating a Virtual Application Package. As such, it is recommended that the following programs should be shut down before you begin sequencing. Some of these programs include:
  + - Windows Defender
    - Antivirus software
    - Disk defragmentation software
    - Windows Search
    - Microsoft Update
    - Any open Windows Explorer session

**Note**: The Sequencer workstation should be fully scanned for viruses and malware and then the antivirus and antimalware software should be disabled before creating a snapshot image of the Sequencer workstation.

# New features of App-V 4.6 SP1 Sequencer

Powerful new features have been added to streamline the sequencing workflow, making sequencer faster for the sequencing engineer, as well as to increase the likelihood of developing successful sequences when dealing with more complex application suites.

## Changes to the Sequencer in 4.6 SP1

Following lists the changes to the App-V 4.6 Sequencer that will be detailed in the section below.

* Improved user interface to simplify sequencing process
* Package Accelerators
* Settings Templates
* Built-in diagnostics
* Command Line Sequencer enhancements

### Improved User Interface

The enhanced wizard will guide you through creating new packages and modifying existing packages. When creating a new package, the wizard will create a default package root folder based on your package name, and will launch the application installer after you specify it. The configure software wizard page allows you to open applications and prepare them for first-time use by entering licensing information or accepting EULAs. You can create a basic package or extend the wizard to make optional modifications, such as configuring shortcuts and File Type Associations (FTAs), or optimizing for streaming.

There are three main application types in the Create New Package wizard:

* **Standard Application** – The default application type. Most applications should be sequenced using this option.
* **Add-ons or Plug-ins** – Choose this option if you are sequencing an add-on to another application that is either traditionally installed on the client machine, or that will be another App-V package that will be linked through Dynamic Suite Composition.
* **Middleware** – This option is for sequencing middleware or frameworks that will be configured as dependencies for other virtual applications through Dynamic Suite Composition.

There are three main options in the Modify Package wizard:

* **Update Application in Existing Package** – Select this option if you are applying an update or patch to an existing package.
* **Edit Package** – This option will open your package in the console to let you modify deployment options and package contents.
* **Add New Application** – Select this option if you are adding an additional application to an existing package.

### Package Accelerators

Microsoft Application Virtualization Package Accelerators allow you to convert traditional applications into virtual application packages without manually sequencing an installation. App-V Package Accelerators are created by software vendors or other experienced packagers who have already sequenced complex applications. App-V Package Accelerators save sequencing time and help accelerate your App-V deployments.

You can use App-V Package Accelerators to automatically sequence large, complex applications. When you apply an App-V Package Accelerator you are not required to manually install an application to create the virtual application package. The App-V Sequencer extracts the required files from the App-V Package Accelerator and associated installation media to create a virtual package without having to monitor the installation of the application.

You can create and save your own App-V Package Accelerators using the App-V Sequencer. After you create App-V Package Accelerators you can share them with other administrators.

### Sequencer Templates

Use templates to save settings that can then be applied when you create new virtual application packages in your environment. This can help streamline the process of creating virtual application packages.

Microsoft App-V project templates save commonly applied settings associated with an existing virtual application package.

App-V project templates differ from App-V Package Accelerators because App-V Package Accelerators are application specific, and App-V project templates can be applied to multiple applications. The following general settings are saved with an App-V project template:

* **Advanced Monitoring Options** – Allow Microsoft Update to run during monitoring, Rebase **.dll’s**
* **Package Deployment Settings** – Protocol, Host Name, Port, Path, Operating Systems, Enforce Security Descriptors, Create MSI, Compress Package.
* **General Options** – Allow use of Windows Installer file (**.msi**), Allow Virtualization of Events, Allow Virtualization of Services, Append Package Version to Filename.
* **Exclusion Items** – Exclusion pattern list.

### Diagnostics (Reporting)

This feature helps identify common, known sequencing issues that prevent successful sequencing as early as possible in the sequencing process. The report also provides information that can enable users to quickly fix the problems.

Use diagnostics to quickly, early, and effectively solve common sequencing issues.

### Command Line Sequencer Enhancements

Creating App-V packages using the command-line interface version of the Sequencer is now easier. Use the argument **/LAUNCHALL** to open each application in order to create a primary feature block for App-V which helps optimize streaming. Use **/LAUNCH:** to designate specific applications to open. Use the /**LAUNCHSCRIPT** argument to automate the launching of specific processes.

Also, you can now make use of the new App-V Package Accelerator and Template features when using the Sequencer command line interface.

# Recommended Best Practices for Sequencing

This section covers Microsoft’s recommended best practices for sequencing applications.

It is recommended that you familiarize yourself with the installation and execution of the application prior to sequencing. Be sure to read all installation instructions associated with the application. It is also recommended that you learn how the application runs and the components of the application that the user will need.

To improve the process of sequencing an application, one should document step-by-step the installation and post-configuration procedures for the application. Step-by-step documentation will ensure that no unnecessary troubleshooting occurs during the sequencing process since no important steps will be skipped. Always document the sequencing process step-by-step creating a “recipe” using a standardized template. Documenting the sequencing process in this way will allow you to hand the recipe to someone else in your organization and have them 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, need to be performed in the Sequencer after the installation?
* What post-installation configuration steps need to take place in the Sequencer?
* How do users commonly use this application immediately after its launch?
* Does this application do something that App-V currently does not support? If so, check the Microsoft Knowledge Base to see if a workaround is available.

Recommended settings, configurations, and process include:

* When sequencing on Windows Vista or Windows 7, ensure that you have User Account Control (UAC) enabled on the sequencing machine if the client machine you are deploying the application to will have UAC enabled as well.
* Use the Comments field in the Sequencer (Abstract Tag) to note any details about the package you may want to include. This will allow you to revisit the sequence later and have a record of this information.
* The installation drive on your Sequencer should match the virtual drive on the client. In some cases this may not be possible (if, say, packaging for multiple organizations), and you may be required to go back and edit the virtual registry or some individual files to point to the correct locations.
* Sequence to a folder in the root of the drive, not to a subdirectory. (‘Q:\MYAPP’ is correct; ‘Q:\’ is incorrect; ‘Q:\Temp\_Junk\MYFOLD’ is incorrect). If the suite has multiple parts, install each application in a subdirectory of the Asset Directory. For example, if a package contains a Line of Business Application along with the Oracle Client, use Q:\AppSuite as the Asset Directory; sequence the LOB application to Q:\AppSuite\LOB; and sequence the Oracle client to Q:\AppSuite\OracleClient.
* 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, 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 if necessary, until the program is in a static state in the Installation Phase. For example, 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. The multiple launches will make sure that any post-installation tasks that are required by the application can be completed (e.g., accepting a license agreement or setting file type associations).
* If deploying via streaming, use 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.
* Disable “Install on First Use.” Some applications have the option to “Install on First Use” for certain components. It is required that none of the components are sequenced with this option. It is necessary to choose either “Run from My Computer” (install this component) or “Not Available” (do not install this component). For application components that will not be used by any of the targeted users, it is recommended that the components not be installed.
* 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.
* Operations made during the “Prepare for Streaming” section will be included in the primary feature block. As a general rule, when building the primary feature block make sure you execute the application’s most common operations (e.g., in Microsoft Word, open a document and misspell a word to include spell check function) so that they are included in the initial streaming of the application and you have an accurate primary feature block. If this is not done, users will 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 primary feature block. Additionally, if you are in an environment where bandwidth is limited, then you want to have an accurate primary feature block so that users are not constantly making calls to the server to download additional files in cache.
* There are online articles relating to sequencing best practices :
  + Microsoft Support: <http://support.microsoft.com/kb/932137>

# 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 finally, 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 are also very general in nature. 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. These applications might require some changes while sequencing to function correctly. In some cases, they may require no changes but have a larger install that takes more time. In rare occurrences you will encounter both. Changes you might encounter in these packages would be on the order of making changes to the registry, altering the .osd file to launch with additional parameters and scripts, or there may be additional applications needed to install together as a suite so they can function together. | **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 3-4 GB in size and may require compression to get the package under the 4GB App-V limit. Other hurdles you may encounter are the application’s reliance on files being in a specific place and functions hard-coded to that install. These applications may require you to manually edit batch and command files to point to resources in the virtual environment. If this is the case, it is highly recommended utilizing a program that can scan multiple files and make several changes at once. You also may 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, before you begin, all the pieces must be in place. 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 1: 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 short list of application functions and limitations of the software.

|  |  |
| --- | --- |
| Limitation | Description |
| Sequenced package over 4GB (SFT has 4GB limit) | * By utilizing compression within the Sequencer it’s possible to take applications that are larger than 4GB and get them below the limit. * However, after compression the application must be smaller than 4GB. * If the application is too large then the Sequencer will not save the application. It will, however, attempt to compress the file. |
| 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 you would include both programs in the same suite, however if this application launches or initiates commands in several applications it may not be feasible for you to include all of the applications in the same suite. * This is especially true if one of the reasons you are 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 made dependent on one another via [Dynamic Suite Composition.](#_Dynamic_Suite_Composition) |
| 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 2: Sequencing Limitations

# Sequence files

When you save your package you will have several files. The following is a list of those files and descriptions of each of them.

|  |  |
| --- | --- |
| **File** | **Description** |
| .SPRJ | Sequencing project with references to OSD, default package settings, list of all parser items, classifications and exclusions. |
| .SFT | Binary file containing all assets and state organized into stream able feature blocks. Optionally a DSFT (Differential SFT). |
| .OSD | Description of an application including environment, dependencies, package location, shell integration, scripts. |
| .ICO | Icons associated with each shortcut or FTA defined in an OSD or Manifest. These are extracted from application resources. |
| .XML Manifest | Publishing parameters for all applications in a package. Includes definitions of shell integration (FTAs, Shortcuts, DDE, etc.). |
| .MSI | Installer to publish and load (“install”) a virtual application package in standalone environments. Embeds all but SFT. |
| .CAB | *Optional:* Package Accelerator file. |
| .SPRT | *Optional:* Sequencer Template file. |

Table 3: Sequencing Output Files

# Sample Sequencing

This section describes some of the key points to remember during sequencing. 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.

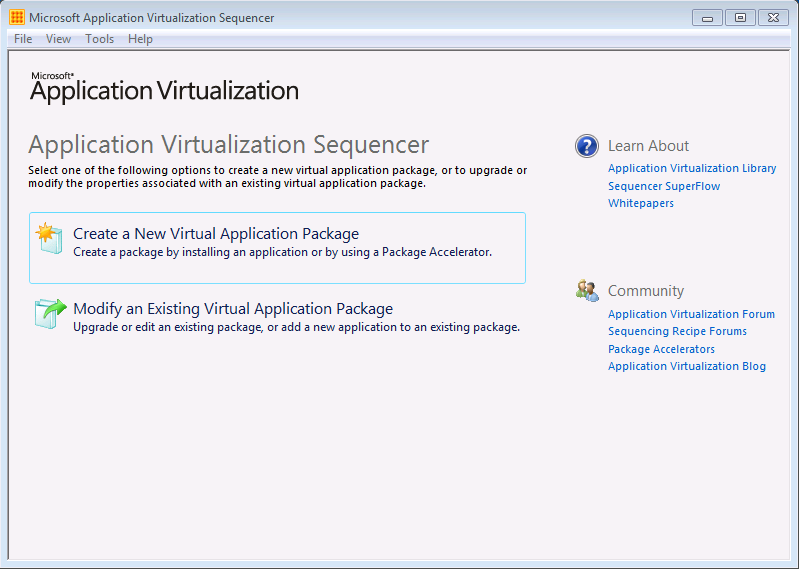


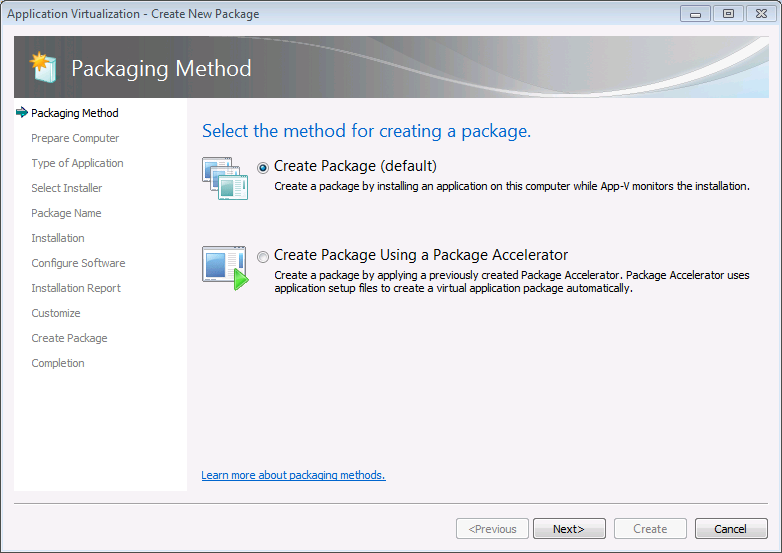
Figure 1: Welcome Screen

|  |  |
| --- | --- |
| 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 4: 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 5: Package Method Component Description

## Prepare Computer

Next, the Sequencer will quickly examine the current operating environment to look for any known process or conditions that are in place (such as if the Sequencer hasn’t been reverted to a clean state after a previous sequencing operation, or there are pending reboots) that might prevent a successful sequencing. Note in the example below that the Sequencer is recommending to stop the Windows Defender service before continuing.

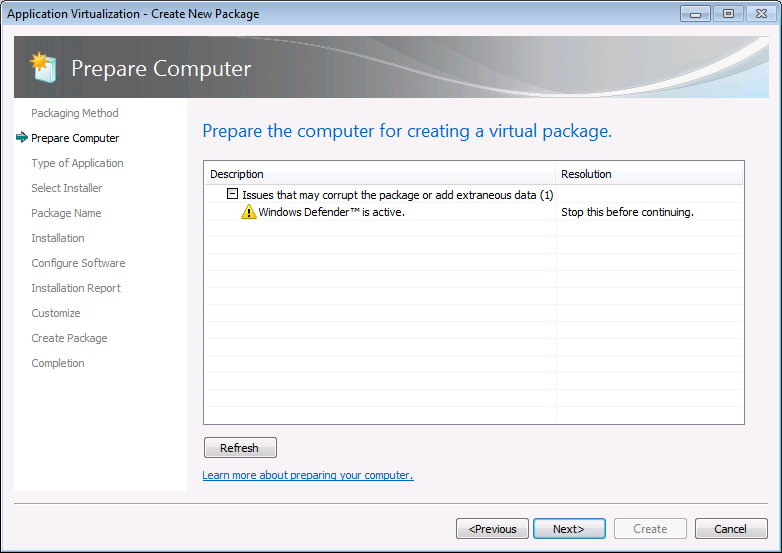


Figure 3: Prepare Computer

## Type of Application

The Sequencer offers choices for the type of application being sequenced in order to determine the steps in the wizard. 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 [Dynamic Suite Composition](#_Dynamic_Suite_Composition) solutions.

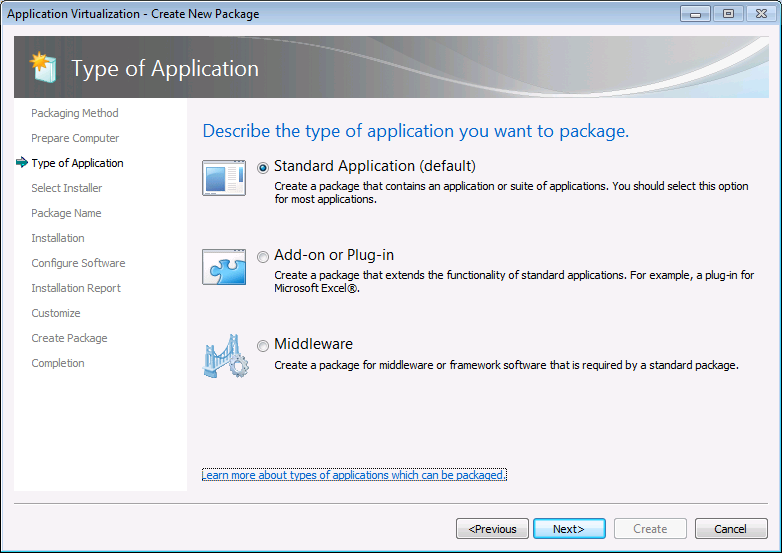


Figure 4: Type of Application

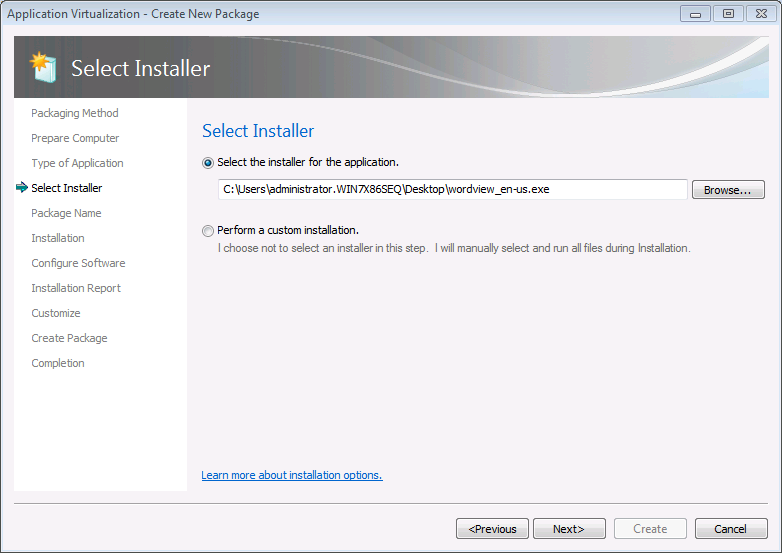
|  |  |
| --- | --- |
| Advanced Options Component | Description |
| Standard Application | Select this option if sequencing a single application or suiting multiple applications into the same virtual application package. |
| Add-on or Plug-in | Select this option if sequencing multiple applications in separate virtual application packages and linking them using Dynamic Suite Composition. This option can also be used when packaging Add-ons or Plug-ins for locally installed applications like Internet Explorer. |
| Middleware | Select this option if sequencing multiple applications in separate virtual application packages and linking them using Dynamic Suite Composition. 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 6: Type of Application Component Descriptions

## Select Installer

Next, 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, you can also select to “Perform a custom installation”. This option causes the sequencing wizard to enter monitoring and then wait for you to finish manually launching any installation tasks. This option is often useful when sequencing applications that may not have an install or setup file such as applications that just 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 for you. |
| Perform a custom installation | Select this option to start monitoring and then manually launch your application installer(s). This option behaves similar to previous versions of the Sequencer. It is useful when you have multiple application installers or when there is no application installer. |

Table 7: Select Installer Component Descriptions

## Package Name

Next, define the package name and primary virtual application directory.

The Primary Virtual Application Directory is the directory off of the root of the Q: drive that will contain all files for the sequence. Long directory names are supported and the Sequencer will automatically fill in this field as you type the Package Name. Select the “Edit” box to manually set the name of the directory.

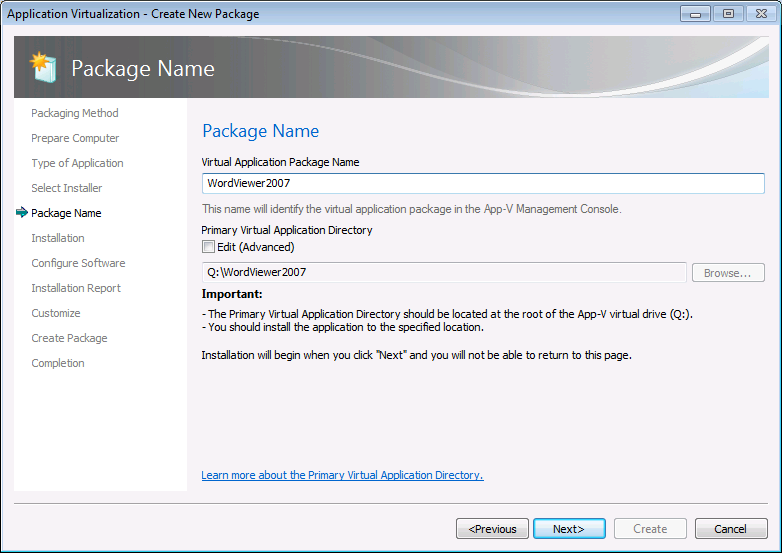


Figure 6: Package Name

|  |  |
| --- | --- |
| Advanced Options Component | Description |
| Virtual Application Package Name | Define a unique name for the virtual application package. |
| Primary Virtual Application Directory | This is the primary directory that the application will reside in once cached on the client in the Q: (default) drive. |

Table 8: Package Name Component Description

## Installation

Next, depending on your earlier choice, the Sequencer will either begin monitoring or launch the installer defined earlier automatically, or you will now manually install the application. You could also click **Run** and select an installer to have the Sequencer launch the installer for you.

Once installation is complete, select the **I am finished installing** check box and click next.

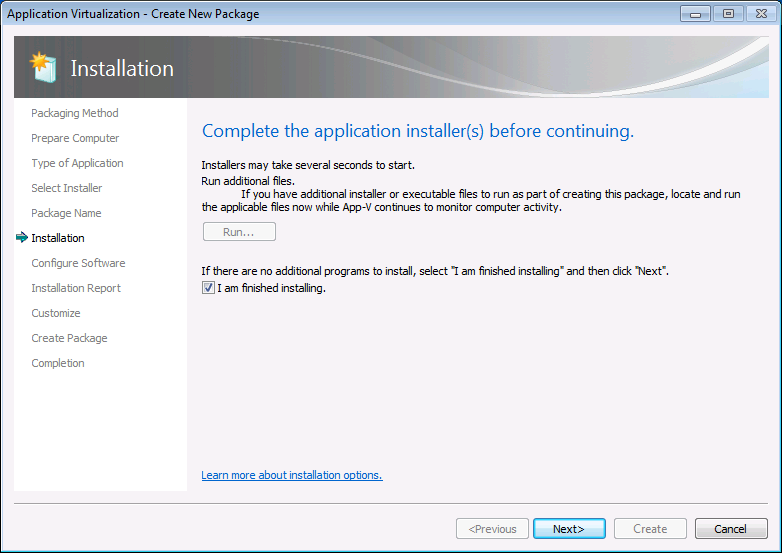


Figure 7: Installation

### Applications that Require a Reboot

The process of App-V sequencing provides an additional benefit over traditional software deployment and installation. Applications that require a reboot to complete the installation will no longer require the reboot when delivered to the client machines. This is because the Sequencer will monitor the operations that need to be completed during reboot and process them prior to saving the package and prior to delivery to client computers.

For applications that give you the choice to reboot, just click “yes.” The Sequencer will record the reboot tasks that the application has configured and cancel the reboot.

If the application forces you to reboot, the Sequencer will record the reboot tasks that the application has configured and cancel the reboot.

Once you check the “I am finished installing” check box and click **Next**, the Sequencer will simulate the reboot. That is the sequencing workstation will not actually reboot. The Sequencer will simply process the reboot tasks that the Sequencer has recorded and apply them to the package.

## 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 and second-run tasks are executed). Also, it is during this execution that you will make any applicable application configuration changes that the Sequencer will capture.

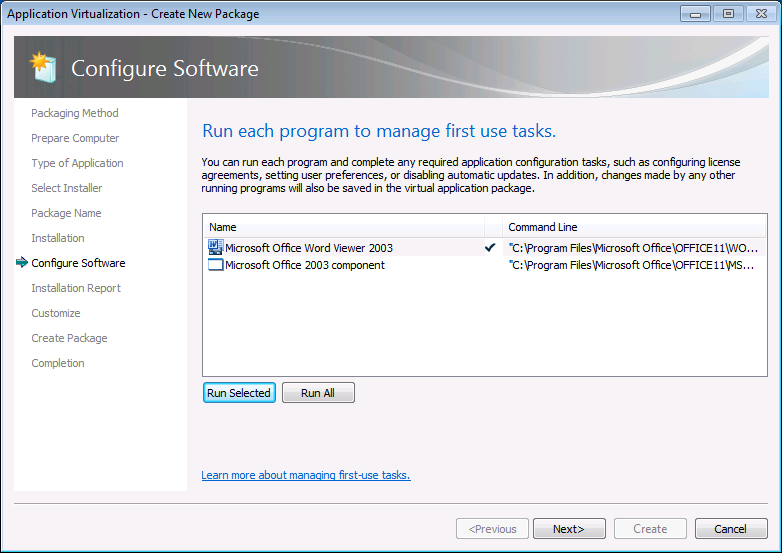


Figure 8: Configure Software

## Installation Report

The Sequencer now detects common sequencing 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 about how to resolve it. 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.

Items in the report include:

* Excluded Files
* Drivers
* COM+ System differences
* SxS Conflicts
* Shell Extensions

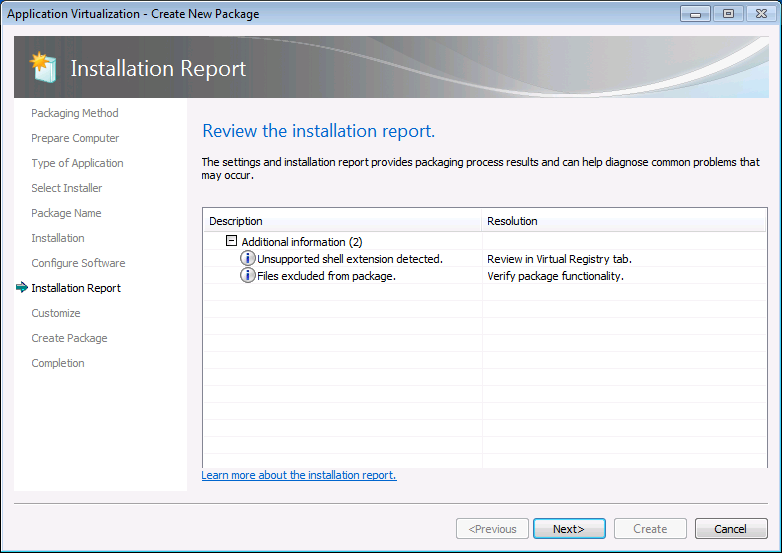


Figure 9: Installation Report

## Customize

Next you have the choice to stop and save the sequence at this point or to further customize the package before saving the package.

Often, there are other steps left to do such as:

* Splitting the package into feature blocks to reduce the streaming requirement and save bandwidth.
* Customizing the name and version numbers of shortcuts and changing their locations and file-type associations.
* Selecting additional client operating systems that will be permitted to receive this package (the default is the current OS).
* Configuring the name, port, and location for the package for distribution via the App-V Management Server (if applicable).
* Further customizing any virtual registry entries.

If you are satisfied that the package you have sequenced will not benefit from any of the above steps, then selecting “Stop Now” may be appropriate. Otherwise, select “Customize.”

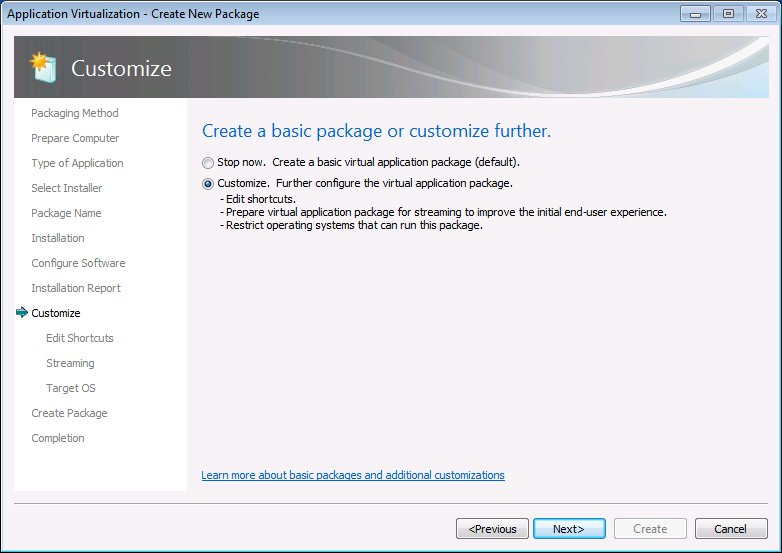


Figure 10: Customize

|  |  |
| --- | --- |
| Advanced Options Component | Description |
| Stop now | Select this option if you are confident that the application as is requires no additional configuration, and does not require the use of Feature Blocks. |
| Customize | Is it recommended to select this option if you are sequencing a larger or more complex application suite and/or if intend to use this package on multiple client operating systems.  This option will continue the sequencing process in the same way as previous versions of the Sequencer, allowing you to:   * Edit and create additional shortcuts * Create Feature Blocks to reduce the amount of traffic used for deployment * Define which operating systems may run the package * Customize registry entries |

Table 9: Customize Component Descriptions

## Edit Shortcuts

Next, the Sequencer will evaluate all of the applications identified during monitoring. Applications may have to be added or removed from this list, based on what is wanted in the final package. Remember, any application in this list will have an associated OSD file created for launching. Also, with web-based applications it is often required to add Internet Explorer as an application because the web-based application will require Internet Explorer to run properly. Each application can be modified to change the name, icon that it uses, file type associations, and locations for shortcuts on destination computers.

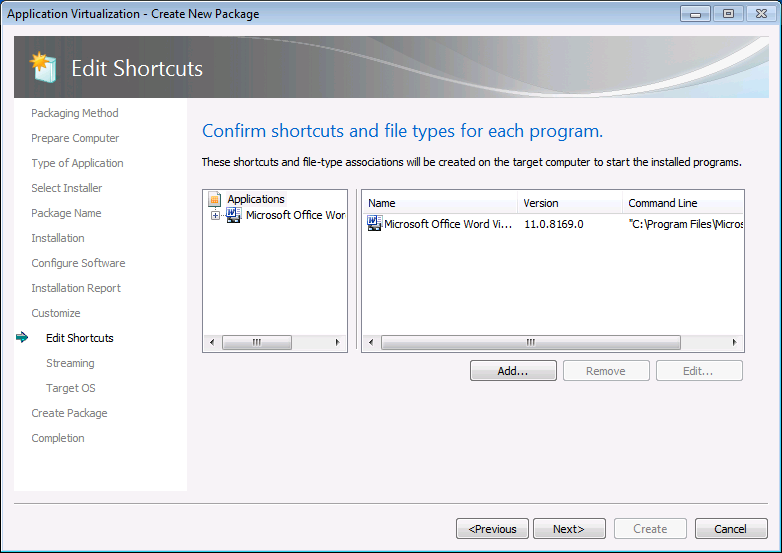


Figure 11: Edit Shortcuts

Clicking the “Edit” button for any shortcut will bring up the options in the following capture (Figure 12) useful for customizing the name and version of the shortcut that will be displayed to the end user as well as the filename for the shortcut (.osd) that will be saved in the sequence.

Clicking the “Add” button will bring up the same option window as the “Edit” button (Figure 12), but the fields will be blank. You would first click the “Browse” button next to the Application Path and select the executable file that the new shortcut will point to. The Icon, Name, Version and OSD file name fields will auto-propagate based on the information within the selected executable. You can then proceed to edit the information to suit your needs.

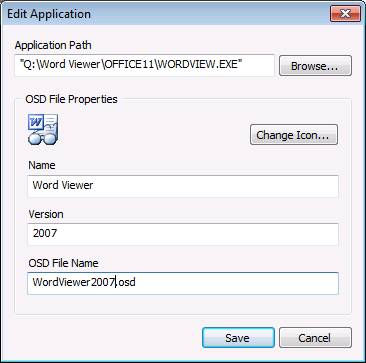


Figure 12: Edit Shortcut Options

|  |  |
| --- | --- |
| Configure Apps Component | Description |
| Edit | At this point you can add or remove icons for applications. Additionally, you can edit the file type associations and the icon locations on the client. |
| Add | Click to create a new application shortcut that was not automatically generated by the Sequencer. |
| Remove | Use this option to remove any shortcut from the package that you do not want presented to your users. |

Table 10: Edit Shortcuts Component Descriptions

## Prepare for Streaming

Feature blocks are designed to optimize the applications for streaming (if applicable), creating a minimum launch threshold to allow larger applications to launch as soon as enough of the package has been downloaded to allow the user to perform his or her typical functions within the application.

This also reduces 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.

Unless you are deploying your virtual application package using System Center Configuration Manager with the Download and Run option or deploying the virtual application packages via the MSI to standalone mode clients, you will be streaming the virtual application package and will likely benefit from splitting your package in to feature blocks.

At the Prepare for Streaming screen, you have the option of splitting your package in to feature blocks.

Select and run each shortcut from the package that you expect the users to execute in their typical day-to-day operations. Then, perform the common tasks that typical users make use of within each particular application in their typical day-to-day operations.

During this process, the Sequencer is monitoring which specific pieces of the packages resources are being executed and including them in Feature Block One. When a user launches the application for the first time, the App-V client will stream and cache just the data within Feature Block One over the network and will launch the application.

Any pieces of the package that was not included in Feature Block One is placed in to Feature Block Two and resides on the server or storage location until specific resources from within Feature Block Two are called by the App-V client. Those pieces are then streamed on-demand and cached on the client.

You also have the option of simply clicking ‘Next’. Doing so without launching any shortcuts will cause the entire virtual application package to be contained within Feature Block One. In this case, the entire contents of the package will be streamed and cached on the App-V client. This is typically done for very small application packages where the idea of streaming the entire package does not cause any network bandwidth concerns.

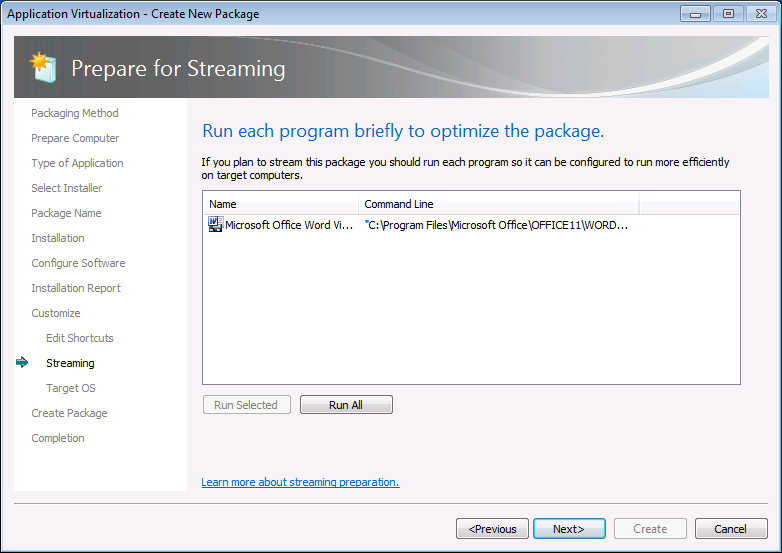


Figure 13: Prepare for Streaming

|  |  |
| --- | --- |
| Launch Applications Component | Description |
| Run Selected | Here you 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 you don’t launch any application 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. |

Table 11: Prepare for Streaming Component Descriptions

## Target OS

At the Target OS screen, you have the option of configuring the package to be deployable to clients running any operating system or to clients running only specific operating systems.

If you have packaged an application that you know will not run on 64-bit systems, you would make sure to leave all 64-bit OS options deselected. This will ensure that the application will never be delivered to those machines.

If you have an application that will only run on Windows 7 and not Windows XP, you would leave Windows XP deselected to ensure the package is never delivered to those machines.

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. In this regard, this screen allows you to target two separate packages of the same application to their appropriate client operating systems.

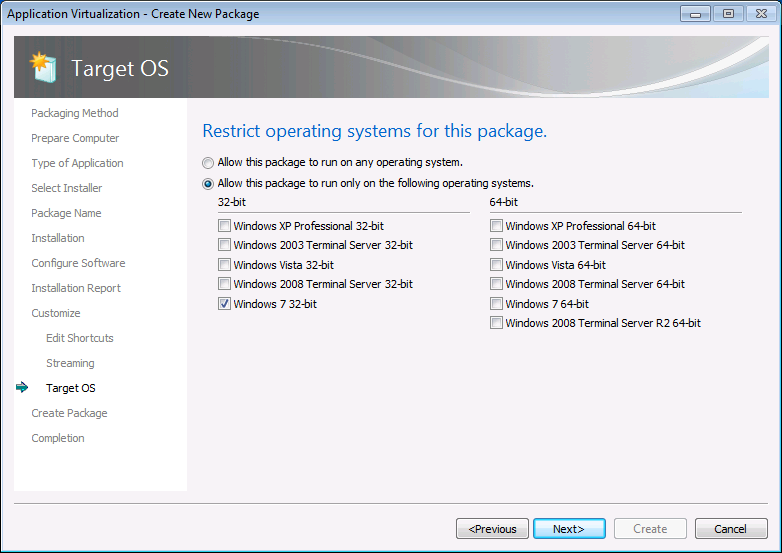


Figure 14: Target OS

## Create Package

At this stage you’re ready to save your package. One easy method of ensuring that the filename and folder you save in are unique is to use the same PACKAGE NAME that you specified in the Package Information wizard.

Be sure to save the package in a new folder named using the package name (example: Desktop\WordViewer2007). Do not save the package in the same folder as the application installer files.

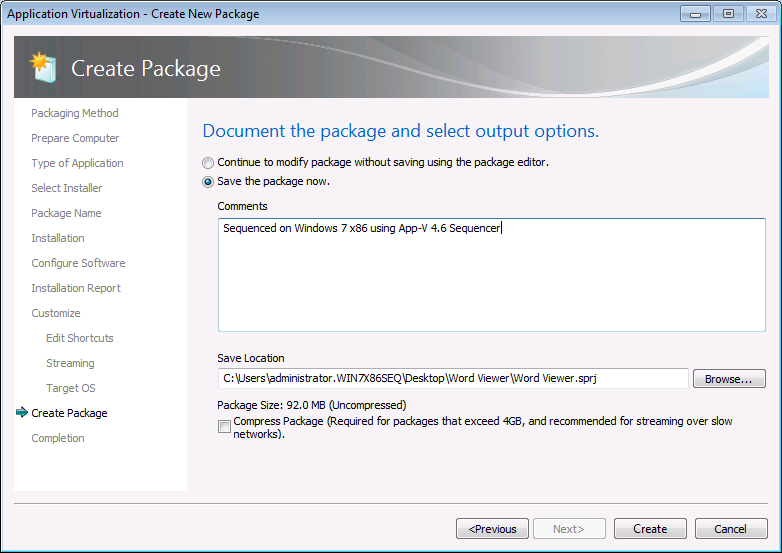


Figure 15: Create Package

|  |  |
| --- | --- |
| Package Deployment | Description |
| Compress Package | The package SFT file can be compressed during sequencing to reduce the amount of traffic that will be streamed on the network when loading the package. The compression will have a one-time impact to the end client the first time the virtual application is launched, as the client PC will need to decompress the SFT and put the data into the client’s cache. The decompression algorithm is CPU-intensive. The default setting is “No Compression.” If you are concerned about the size of the package and the impact it may have on your network you may select to compress the package. |

Table 12: Create Package Component Descriptions

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 about how to resolve it. 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.

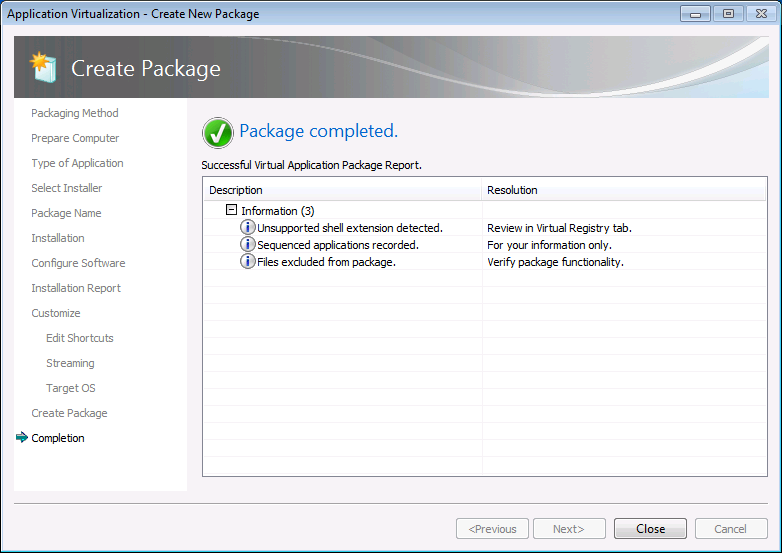


Figure 16: Package Completed

# Creating a Package Accelerator

Once you have completed sequencing an application, you have the option of creating a Package Accelerator. App-V Package Accelerators assist you in packaging applications easily and automatically without having to manually install the application or follow a step-by-step recipe. The wizard will guide you through the process of creating a package using an App-V 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. We recommend that you revert your virtual machine or reimage your physical machine after you apply an App-V Package Accelerator. The wizard will also guide you through the process of creating an App-V Package Accelerator.

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

* .OSD files
* .SPRJ file
* 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

To create an App-V Package Accelerator you first need to have placed on the Sequencer the installation files for the application as well as the existing virtual application package for the application.

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, you will first need to extract the files and present the folder containing the extracted set-up files to the sequencer.
  + Alternatively, at the Installation Files Screen (Figure #18) you can select the option ‘Files installed on local system’. To use this, the application must be already physically installed on the local system.

## Work-flow

* Once you have completed a sequence, click the “Tools” pull-down menu and select “Create a Package Accelerator.”

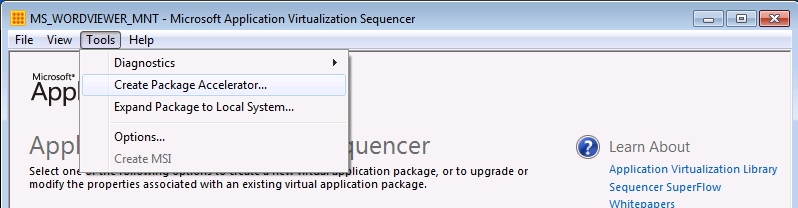


Figure 17: Tools Menu

* At the Select Package screen, browse to and select the .SPRJ 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).

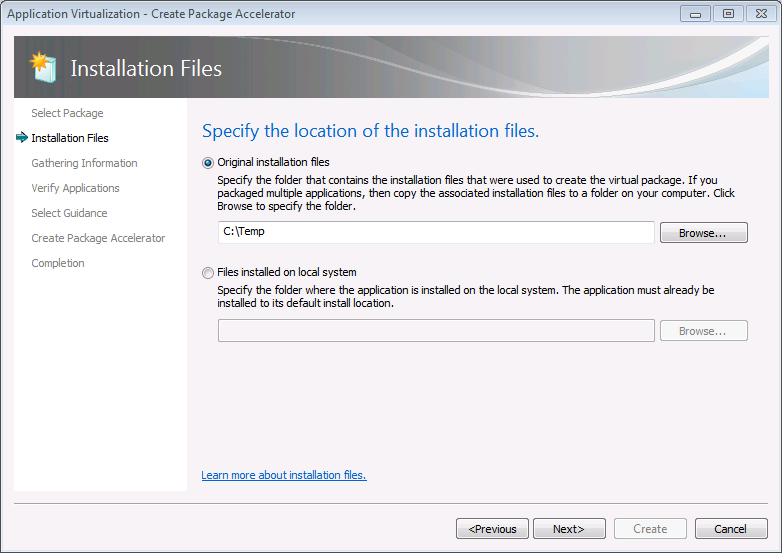


Figure 18: Installation Files

* At the Review Missing Files screen, the Sequencer will identify and report on any files that do not appear where the application expects to find them. Often many of the files may not actually be needed. In that case, click the box labeled “Remove these files.”

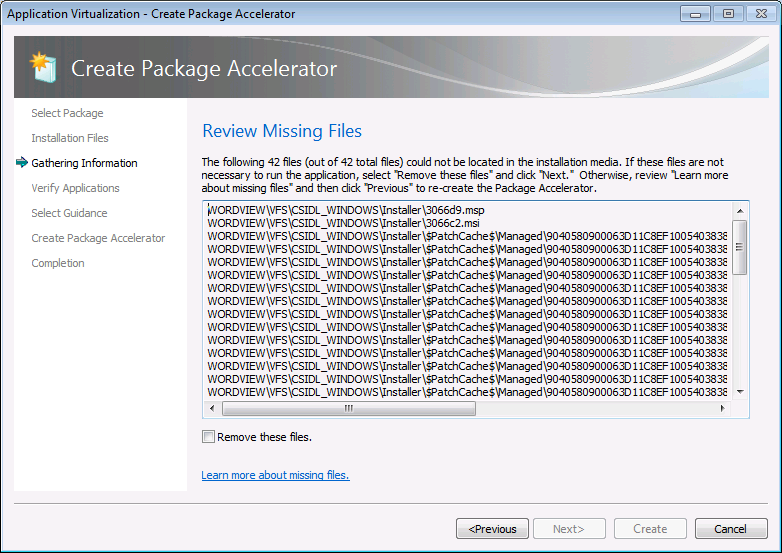


Figure 19: Review Missing Files

* At the Select Files screen, the Sequencer will display all files that have been detected for use in the App-V Package Accelerator. Uncheck the box next to any files that you do not want included in the Package Accelerator.
  + Example: The Sequencer has included a large temporary or cache location containing copies of the setup files that you know you do not need included in the package. Removing files such as this can help reduce the size of the overall package.

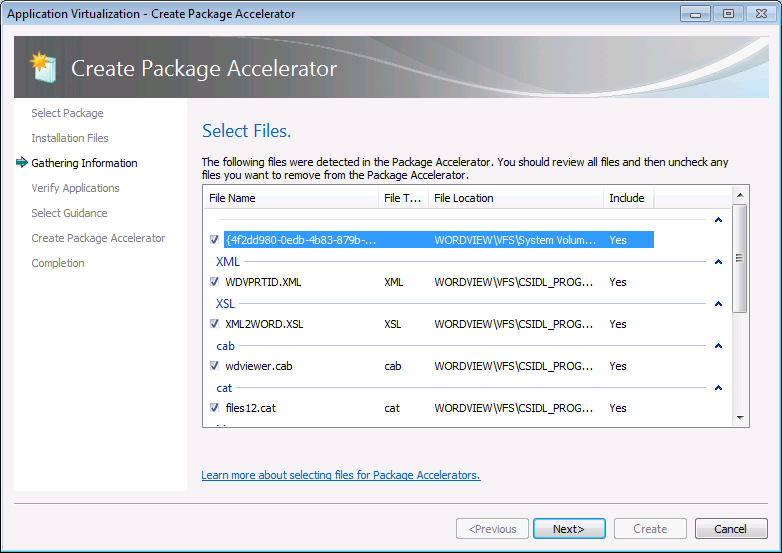


Figure 20: Select Files

* At the Verify Applications screen, the Sequencer will provide a list of applications that were 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 that will be needed.

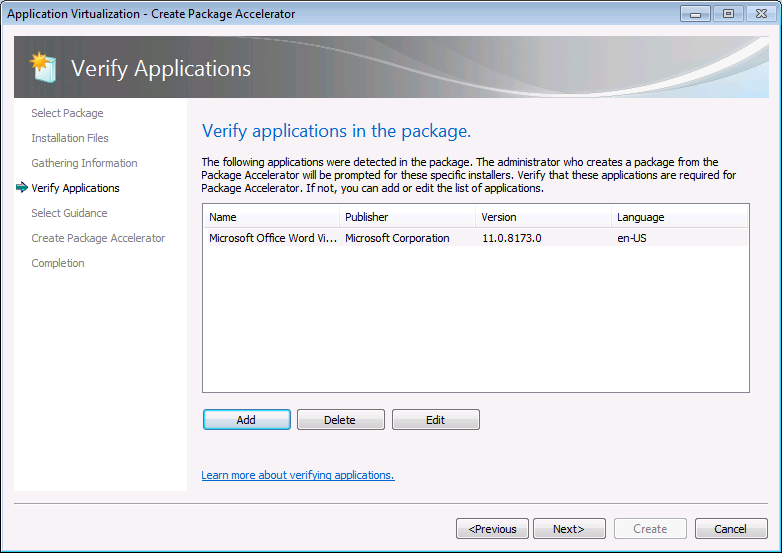


Figure 21: Verify Applications

* At the Select Guidance screen, create a .txt file containing any special instructions needed for the engineer to create the package using the App-V Package Accelerator. Enter the location of this file. 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.

# Create a package using a Package Accelerator

To create a new virtual application package using an App-V Package Accelerator you will first need to have the application installation files and the Package Accelerator (.cab) file copied to the sequencing VM.

## 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 to you 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 you will be making 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 you are certain that no further configuration is necessary you may select to ‘Skip this step’ and proceed directly to saving the package.
9. At the Run Each Program to manage first-use tasks screen, you can choose to run all or run specific applications to clear any first-use tasks as well as further customize the application.
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.

# 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.

There are three different types of Application Upgrades that will be presented to you 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. If you select this option you should have the associated installation files saved locally to the computer running the App-V Sequencer. You must also have access to the location where the package that contains the application or program you want to modify is saved.

The wizard will walk you 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. To edit a package, you must have access to the location where the virtual application package is saved. The following list displays the package properties that can be updated if you select 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.

• Add file type associations.

• Rename shortcuts.

• 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 the additional applications. For example you can add Microsoft Excel to an existing Microsoft Office virtual application package. To add a new application you must have access to the location where the virtual application package is saved.

The wizard will walk you 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 to replace the 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 .sft file name. (example: WordViewer2007\_2.sft) The number corresponds with the version (iteration) of the application package. You would apply this new version to your distribution method (varies) to upgrade or replace the version currently deployed to the clients.

## Updating a Package that will Run at the Same Time as 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.”
* Select the “File” pull-down menu then “Save As” to save the package (Figure 22). 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 you specified. You would apply this new individual package to your distribution method (varies) to deploy for operation in tandem with the previous version on the clients.

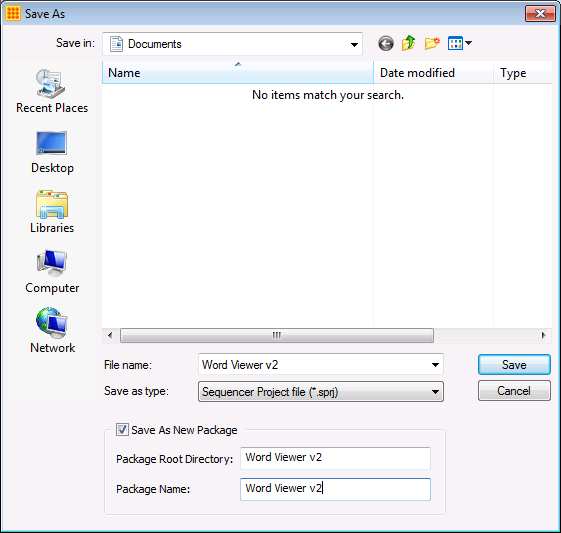


Figure 22: Save As New Package

# Dynamic Suite Composition

Dynamic Suite Composition (DSC) in App-V enables you to define an application as being dependent on another application, such as middleware or a plug-in. This enables the application to interact with the middleware or plug-in in the virtual environment, where normally this is prevented. This is useful because a secondary application package can be used with several other applications, referred to as the primary applications, enabling each primary application to reference the same secondary package.

You can use Dynamic Suite Composition when sequencing 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). If each application that used these dependent components needed to be sequenced, including the components, updates to those components would require re-sequencing all the primary applications. By sequencing the primary applications without those components and then sequencing the middleware or plug-in as a secondary package, only the secondary package needs to be updated.

## Advantages

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

**Note: The secondary application can be streamed in the normal way and does not need to be fully cached to run.**

A primary package can have more than one secondary package. However, only one level of dependency is supported, so you cannot define a secondary package as dependent on another secondary package. Also, the secondary application can only be middleware or a plug-in and cannot be another full software product.

If you plan to make several primary applications dependent on a single middleware product, be sure to test this configuration to determine the potential effect on system performance before deploying it.

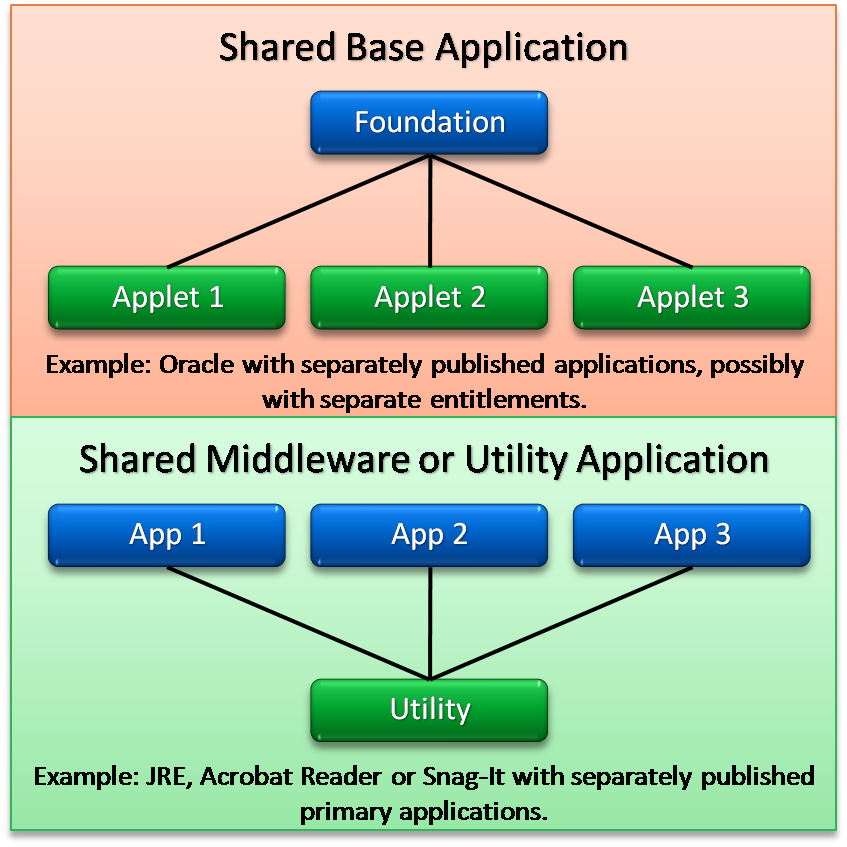


Figure 23: DSC Modes

## Workflow

The App-V 4.6 SP1 Sequencer user interface has been designed to include guided workflows for working with Dynamic Suite Composition.

When selecting Create a New Virtual Application Package from the Welcome Screen, you are then presented a new screen which provides options for the type of package you are creating. The “Add-on or Plug-In” and “Middleware” options are options to create Dynamic Suite Compositions using the Sequencer user interface.

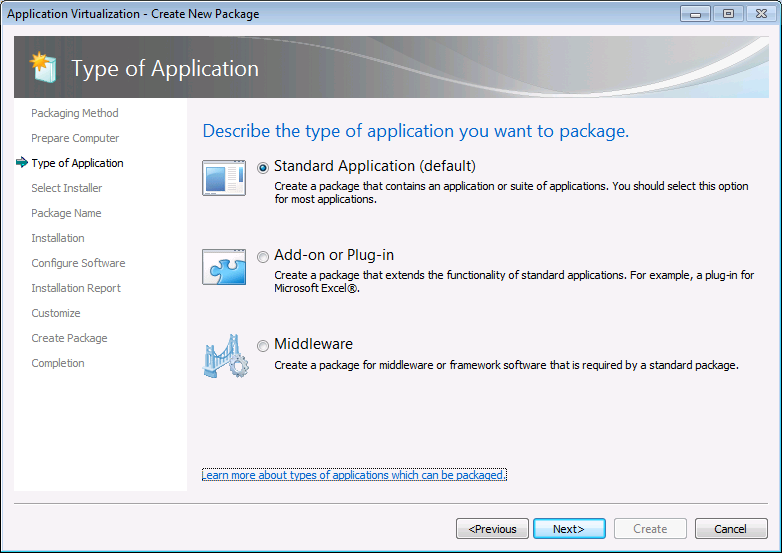


Figure 24: Type of Application

In previous versions of the sequencer, when sequencing for Dynamic Suite Composition, you needed to physically install the previously sequenced primary app on to the Sequencer before sequencing the secondary applications. App-V 4.6 SP1 introduces the ability to use the previously packaged SFT, expanding it to the Sequencer system, saving time and effort and improving the reliability of creating Dynamic Suite Composition packages. The process of creating Add-on, Plug-in or Middleware packages for Dynamic Suite Compositionis detailed in the following sections.

## 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 work flow for plug-ins is:

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.
3. Expand the Primary App package back to the clean sequencing virtual machine.
   1. Copy the primary app package back to the sequencing virtual machine.
   2. Open the Sequencer and select the **Tools | Expand package to local system**
      1. Select the primary application package.
      2. **Note:** This process expands (or de-virtualizes) the primary application package physically on to the local system so the necessary bits are available to the add-on or plug-in being virtualized.
4. 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. At the Select Primary screen, browse to the main executable for the parent program.
      1. Example: You are sequencing a plug-in for Microsoft Excel. You would browse to and select the Excel.exe file that you’ve already expanded back to the Q: drive. This will cause the sequencer to create a new .OSD file for Excel.
   6. At the Package Name screen, create a unique name for the add-on or plug-in package.
   7. At the Installation screen, install the plug-in. When installation is complete, back in the sequencer select the ‘I am finished installing’ check box and click ‘Next.’
   8. Complete the package.
5. Link the dependencies between the two packages
   1. Use either the Dynamic Suite Composition tool (Section 12.5) or manually modify the XML code in the relevant .OSD files to create the dependency to the plugin-package (Section 12.6).

## Middleware

When creating a separate package to contain middleware to be available to other primary applications, for example, Java 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 8 of this document.
2. Revert the sequencing virtual machine.
3. Expand the middleware app package back to the clean sequencing virtual machine.
   1. Copy the middleware app package back to the sequencing virtual machine.
   2. Open the Sequencer and select the **Tools | Expand package to local system**
      1. Select the middleware application package.
      2. **Note:** This process expands (or de-virtualizes) the middleware application package physically on to the local system so the necessary bits are available to the primary application being virtualized.
4. Sequence the Primary 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 ‘Standard Application (default).’
   4. At Select Installer either select the installer file for the Primary Application or select ‘Perform a custom installation’ just as in Section 8.
   5. At the Package Name screen, create a unique name for the Primary Application.
   6. At the Installation screen, install the primary app. When installation is complete, back in the sequencer select the ‘I am finished installing’ check box and click ‘Next.’
   7. Complete the package.
5. Link the dependencies between the two packages
   1. Use either the Dynamic Suite Composition tool (Section 12.5) or manually modify the XML code in the relevant .OSD files to create the dependency to the plugin-package (Section 12.6).

## Dynamic Suite Composition Tool

The Dynamic Suite Composition tool provides a GUI interface that makes it easier to configure packages to take advantage of the dynamic suiting feature. This tool also eliminates the possibility of incorrect editing of the OSD files required to take advantage of dynamic suiting. The Dynamic Suite Composition tool also modifies the MSI files created as part of the package to reflect the changes to the embedded OSD files that are necessary for Dynamic Suite Composition. The tool can be installed and used from any machine. The following outlines the steps required to use the Dynamic Suite Composition tool.

**Note:** The Dynamic Suite Composition tool requires properly sequenced packages that will support dynamic suiting. The tool facilitates editing the OSD files to support dynamic suiting only.

1. Download the Dynamic Suite Composition tool from: <http://www.microsoft.com/downloads/details.aspx?familyid=DAA898DF-455F-438A-AA2A-421F05894098&displaylang=en>.
2. Run the self-extracting EXE and select a destination directory for the binaries.

**NOTE: The Dynamic Suite Composition tool doesn’t require installation; it is run directly from the directory where it was extracted.**

1. Open the Dynamic Suite Composition tool.

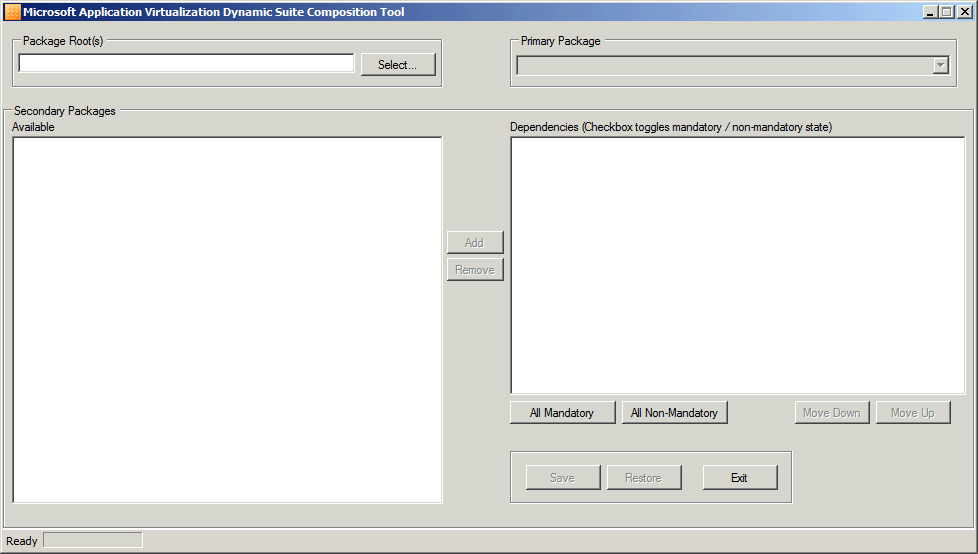


Figure 25: Dynamic Suite Composition Tool

1. Click on **Select** to choose the directory where the App-V packages for dynamic suiting exist.
2. After selecting the package directory, select the **Primary Package** from the drop-down list.

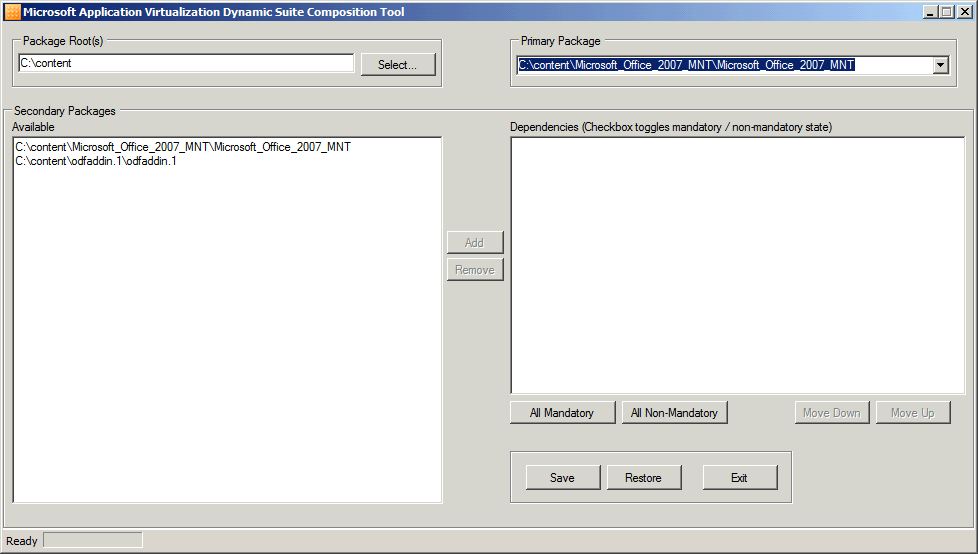


Figure 26: Dynamic Suite Composition Tool

1. Select and **Add** the Secondary Package(s) from **Available** list.

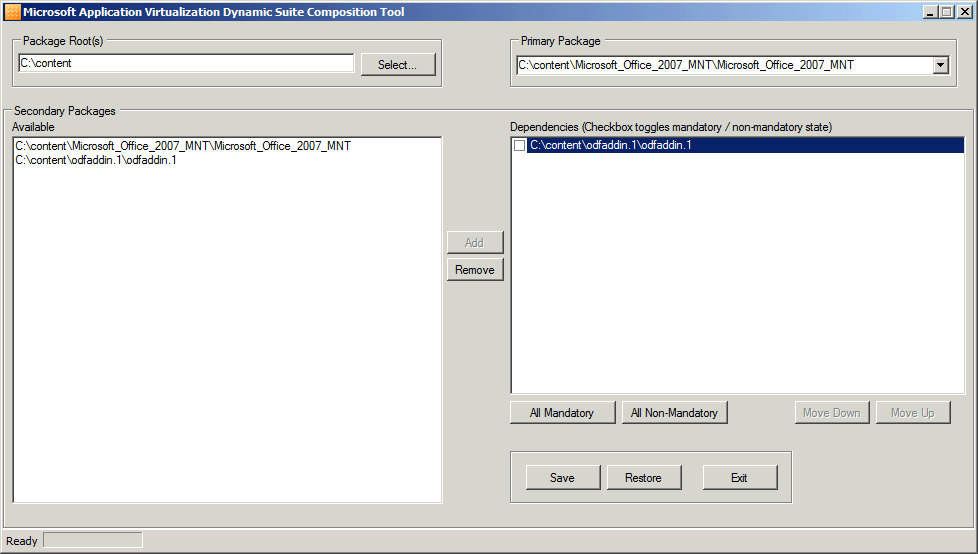


Figure 27: Dynamic Suite Composition Tool

1. Select the **Checkbox** to the left of any **Dependencies** that are mandatory or use the **All Mandatory / All Non-Mandatory** buttons to toggle the setting for multiple dependencies at one time.
2. Select **Save** to apply the changes to the OSD files to support dynamic suiting.

**Note: The Dynamic Suite Composition tool creates backup copies of the original OSD files with the extension BAK. These files can be restored using the tool by selecting the primary application and selecting Restore.**

## Manually Defining the Dependency in the Primary Package

In both above examples, the primary package will need to be modified to incorporate the secondary package(s) for Dynamic Suite Composition.

1. On the server, open the OSD file of the secondary package for editing. (It is advisable to use an XML editor to make changes to the OSD file; however, you can use Notepad as an alternative.)
2. Copy the **CODEBASE HREF** line from that file.
3. Open the OSD file of the primary package for editing.
4. Insert the **<DEPENDENCIES>** tag after the close of **</ENVLIST>** tag at the end of the **<VIRTUALENV>** section just before the **</VIRTUALENV>** tag.
5. Paste the **CODEBASE HREF** line from the secondary package after the **<DEPENDENCIES>** tag you just created.
6. If the secondary package is a mandatory package, meaning that it must be launched before the primary package is launched, add the **MANDATORY=”TRUE”** property inside the **CODEBASE** tag. If it’s not mandatory, the property can be omitted.
7. Close the **<DEPENDENCIES>** tag by inserting the following:

**</DEPENDENCIES>**

1. Review the changes made to the OSD file, and then save and close the file. The following example shows how the added section should appear. The tag values shown here are for example only.

**<VIRTUALENV>**

**<ENVLIST>**

**…**

**</ENVLIST>**

**<DEPENDENCIES>**

**<CODEBASE HREF="rtsp://virt\_apps/package.1/package.1.sft" GUID="D54C80FA-9DFF-459D-AA33-DD852C9FBFBA" SYSGUARDFILE="package.1\osguard.cp"/>**

**<CODEBASE HREF="rtsp://sample\_apps/package.2/sample.sft" GUID="D54C80FA-9DFF-459D-AA33-DD852C9FBFBA" SYSGUARDFILE="package.2\osguard.cp" MANDATORY="TRUE" />**

**</DEPENDENCIES>**

**</VIRTUALENV>**

**Note:** If virtual applications are deployed via MSI installer, you must use the Dynamic Suite Composition tool to link the packages as it also modifies the contents of the MSI files for the primary application.

# Sequencer Templates

The App-V 4.6 SP1 Sequencer provides the ability to create sequencer template files.

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.

For creating 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.

To create a template also containing additional options such as Package Deployment Settings and Advanced Monitoring Options, the engineer will select **File | Save As Template** upon completion of a particular sequencing where the engineer has specified these options.

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

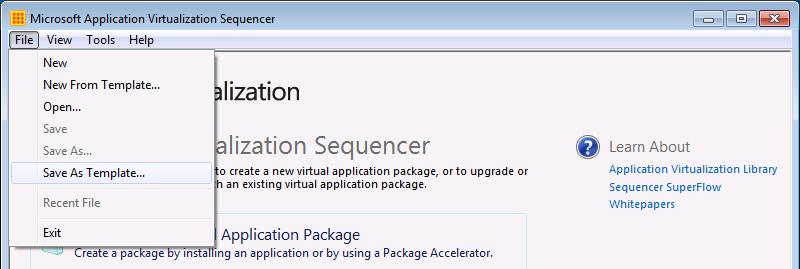


Figure 28: Save as Template

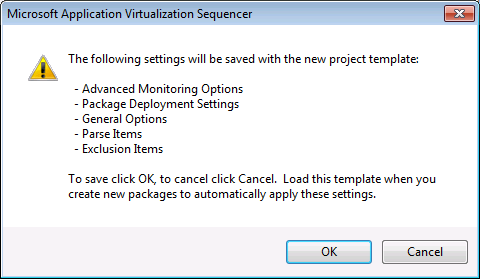


Figure 29: Template Information

# Advanced OSD Scripting

In some instances it may be necessary to make modifications to the application to get it to function properly in the App-V environment. One of the most common methods is making modifications to the OSD file to allow certain actions to be performed at a specified time during the execution of the application.

Scripting in the OSD file is an invaluable tool when sequencing / publishing applications. These scripts can be used to setup or alter the virtual environment before an application executes, or 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.

It is important to understand the OSD files, and their syntax, prior to using custom scripting. Only experienced sequencing engineers should make custom modifications to the OSD file, and you should always create a backup copy of the original OSD file prior to making any modifications. Placing an incorrect syntax in an OSD file could cause the application to not work properly, or can even prevent the application from appearing at all on the clients.

**OSD file**: Provides information necessary to launch the application, including the name of the Virtual Application Server from which to stream the application, and any Dynamic Suite Composition dependencies. Each application requires an OSD file. The OSD file is written in an XML format. The major sections are detailed below:

* **<SOFTPKG>:** This line displays the package’s Globally Unique Identifier (GUID), name and version. Each of these fields must be unique across all published applications.
* **<IMPLEMENTATION>:** This section tells the Application Virtualization Client "how" to run the virtual application, including what protocol to use when streaming the application, what server to get the application from, whether or not to "terminate" any child processes when the application closes, etc.
* **<DEPENDENCY>:** This section lists any scripts that must be run prior to launching this application. It also specifies whether a specific version of the App-V Client must be installed on the client to run this application.
* **<PACKAGE NAME>:** This line lists the name of the package, as specified during sequencing.
* **<ABSTRACT>:** This section displays any comments that were listed during sequencing. You should list as much information here as necessary to describe the sequencing environment.
* **<MGMT\_SHORTCUTLIST>:** This section displays where the shortcuts will be placed on the client. For example, to place a shortcut on the Start\Programs\Virtual\_Applications menu, this would display as "%CSIDL\_PROGRAMS%\Virtual Applications."
* **<MGT\_FILEASSOCIATIONS>:** This section lists all of File Type Associations that will be registered on the client, so an application can be launched by double-clicking a file.

A sample OSD file:

Before we talk about scripting here is a generalized look at a typical OSD file.

<**SOFTPKG** GUID="*application GUID*" NAME="*application name*" VERSION="*version string*">

<**IMPLEMENTATION**>

<**CODEBASE** HREF="*location*“ GUID="*package*" FILENAME="*application*“

PARAMETERS=“*parameters*” SYSGUARDFILE="*state file*"/>

<OS VALUE="*platform*"/>...<OS VALUE="*platform*"/>

<WORKINGDIR>*optional working directory*</WORKINGDIR>

<**VIRTUALENV** *virtual environment configuration options*>

<ENVLIST>*optional variables*</ENVLIST> <REGISTRY>*optional values*</REGISTRY>

<POLICIES>*optional policies*</POLICIES>

<DEPENDENCIES>

<CODEBASE ... MANDATORY="TRUE|FALSE"/>...<CODEBASE ... MANDATORY="TRUE|FALSE"/>

</DEPENDENCIES>

</**VIRTUALENV**>

</**IMPLEMENTATION**>

<**DEPENDENCY**>

<CLIENTVERSION VERSION="4.5.0.0"/> <SCRIPT>*optional scripts*</SCRIPT>

</**DEPENDENCY**>

<**MGMT\_SHORTCUTLIST**><SHORTCUT *definition*>…<SHORTCUT …></**MGMT\_SHORTCUTLIST**>

<**MGMT\_FILEASSOCIATIONS**>

<PROGIDLIST>*shell integration*</PROGIDLIST><FILEEXTENSIONLIST>*FTAs*</FILEEXTENSIONLIST>

</**MGMT\_FILEASSOCIATIONS**>

</SOFTPKG>

## OSD Scripts

All scripts must be placed between the <DEPENDENCY> tags within the OSD files. You can choose to refer to an existing script by its file name or you can enter the exact syntax of the commands in the script section.

When developing scripts, the first step is to decide when the script will run. Using the **SCRIPT TIMING** and **EVENT** tags, the script can be executed at different times during the virtual application launch.

* **Pre Stream:** After the user launches the application, but prior to the application blocks streaming to the client workstation and the virtual environment being set-up. An example of a Pre-Stream script might be starting a necessary VPN connection to connect to a server that the application requires.
* **Post Stream:** After the user launches the application, and after the application blocks stream to the client workstation, but prior to the virtual environment being set-up. An example of a Post-Stream script might be to install a dependent component on the host workstation.
* **Pre Launch:** After the user launches the application, and after the bits stream to the client workstation, and after the virtual environment has been set-up. Whatever commands are called in a Pre-Launch script occur within the virtual environment. An example of a Pre-Launch script might be to ensure a file exists or a registry key gets added to the virtual environment.
* **Post Launch:** After the application launches, but prior to the user getting access to the application’s interface. An example of a Post-Launch script might be calling a script to synchronize a folder inside the virtual environment with one that is outside the virtual environment.
* **Post Shutdown:** After the virtual application has been closed. An example of a Post-Shutdown script might be to reset the workstation back to a clean state to remove any extraneous files that the application itself may have dropped onto the workstation.

**Note:** There is no Pre-Shutdown script as it would be impossible to know the exact moment before the user closes the application.

Next, the script must define where to run the script. The OSD script can be run in one of two locations: either inside the virtual environment or outside, or on the local operating system outside the virtual application virtual environment. The **PROTECT** tag is the only XML tag that defines where the script will run, either inside the virtual environment or outside. The two settings for the PROTECT tag are described below:

* **TRUE**: Allows the script to run within the protected environment. This script placement can be very useful for troubleshooting.
* **FALSE**: Allows the script to run outside the virtual environment. This script placement can be very effective when files need to be copied locally to a client but there is no need to penetrate the virtual environment.

**<SCRIPT TIMING=“PRE” EVENT=“LAUNCH” PROTECT=”TRUE”>**

The last two attributes that affect the script operation are **TIMEOUT** and **WAIT**. These two attributes define how much time the script will need to wait to completely execute. The TIMEOUT attribute, in seconds, defines how long the client will wait. A value of “0” defines the client will wait indefinitely. The WAIT attribute is still supported in version 4.6, for backward compatibility. A value of TRUE defines that the client should wait for the script to complete before starting the next step in the script. A value of FALSE defines that the client can continue onto the next step without waiting for the current step to complete.

**<SCRIPT TIMING=**“**PRE” EVENT=**“**LAUNCH” WAIT=”TRUE” TIMEOUT=”0”>**

**The following sequenced list describes the full application launch and shutdown and when the script events occur. The list also shows whether the virtual environment is loaded and whether or not the script could access the virtual environment.**

1. Virtual Application Launch / Shutdown Progression
   1. User clicks on program shortcut for a virtual application.
   2. **Pre-Stream script runs (outside Virtual Environment).**
   3. Page streams FB1.
   4. **Post-Stream script runs (outside Virtual Environment).**
   5. Virtual environment is loaded.
   6. **Pre-Launch script runs (inside or outside Virtual Environment**).
   7. The virtual application launches.
   8. **Post-Launch script runs (inside or outside Virtual Environment).**
   9. If background streaming is enabled for the application,   
      FB2 will stream into the client cache in the background.
   10. User exits the virtual application.
   11. The virtual environment is unloaded.
   12. **Post-shutdown script runs (outside Virtual Environment).**

**Note:** The client does not support a **SHUTDOWN** event with Timing of **PRE**.

**Note:** Scripts can be written in any language but their language must be installed locally on the client machine. For example, if they are using PERL, the PERL runtime must be locally installed on the client machine.

## OSD: SCRIPTBODY vs. HREF

There are two kinds of scripts, **<SCRIPTBODY>** and **<HREF>,** which are described in more detail:

1. HREF – Normally used when executing a single command.
2. SCRPITBODY – Used when needing more complex operations and/or multiple commands to be run.

|  |  |
| --- | --- |
| <SCRIPTBODY> | <HREF> |
| Can be used to call an executable by searching for the executable in the client’s Path statement. | **Can be used to call an executable by searching for the executable in the client’s Path statement.** |
| Uses the “\” as an escape character and as such requires an extra “\” to pass the actual character. | **Does not use the “\” as an escape character.** |
| Can be used to pass non executable commands such as the DOS make directory (md). This is actually a part of the command.com. | **Cannot be used to pass non-executable commands such as the DOS make directory (md).** |
| When used, the contents of the script are copied to a temporary .bat file on the client’s drive. | **Does not create a temporary .bat file but instead executes the contents of the script directly on the client.** |
| When used a DOS window appears on the screen during the execution of the script’s .bat file. | **No DOS window appears on the screen unless the process being called is a console application.** |
| <SCRIPTBODY> | **<HREF>** |
| Can be used to call an executable by searching for the executable in the client’s Path statement. | **Can be used to call an executable by searching for the executable in the client’s Path statement.** |

Table 13: Scriptbody vs. Href

## <SCRIPTBODY> Examples

**Example 1 – ABORTRESULT and SUCCESSRESULT**

If ABORTRESULT is specified, the client will check the script's exit code when it completes. If SUCCESSRESULT is specified (and if ABORTRESULT is not or it does not match the script's exit code), the client will check the script's exit code to see if it matches the value in SUCCESSRESULT.

<DEPENDENCY> <SCRIPT TIMING="PRE" EVENT="LAUNCH" PROTECT="TRUE" WAIT="TRUE" TIMEOUT="10" **SUCCESSRESULT**="1" **ABORTRESULT**="0">

<SCRIPTBODY>@ if not %COMPUTERNAME% == “SpecialComputer” exit 1 @ exit 0

</SCRIPTBODY>

</SCRIPT>

</DEPENDENCY>

A more detailed explanation of these commands can be found at:

<http://support.microsoft.com/kb/930973/en-us>

**Example 2 – Pre-Launching Command Prompt**

<DEPENDENCY>

<SCRIPT TIMING="PRE" EVENT="LAUNCH" WAIT="TRUE" PROTECT="TRUE">

**<SCRIPTBODY>cmd.exe</SCRIPTBODY>**

</SCRIPT>

</DEPENDENCY>

**Example 3 – Embedding Batch File Commands within OSD File**

<DEPENDENCY>

<SCRIPT EVENT="LAUNCH" TIMING="PRE" PROTECT="TRUE" WAIT="TRUE">

**<SCRIPTBODY>**

**net use k: \\\\w2k-pdc\\netlogon\n**

**CALL k:\\usr-w2k.cmd\n**

**\\\\sfc-App-V\\shr\editini.exe c:\\word\\word.ini "File Locations" TempPath c:\\temp\n**

**</SCRIPTBODY>**

</SCRIPT>

</DEPENDENCY>

**Example 4 – Pre-Launching EXE from within the Sequence**

<DEPENDENCY>

<SCRIPT TIMING="PRE" EVENT="LAUNCH" WAIT="TRUE" PROTECT="TRUE">

**<SCRIPTBODY>%SFT\_MNT%\\OfficeXP\\Office10\\proflwiz.exe</SCRIPTBODY>**

</SCRIPT>

</DEPENDENCY>

**Example 5 – Pre-Launching a Data File (e.g. AVI file) from within the Sequence**

<DEPENDENCY>

<SCRIPT TIMING="PRE" EVENT="LAUNCH" WAIT="TRUE" PROTECT="TRUE">

**<SCRIPTBODY>%SFT\_MNT%\\App\\Data\\Sequencing.avi</SCRIPTBODY>**

</SCRIPT>

</DEPENDENCY>

**NOTE:** In this example, the sequencing.avi launches the locally installed Media Player.

**NOTE:** A data file (e.g. sequencing.avi) cannot be specified within the FILENAME entry of the CODEBASE tag as shown below.

<CODEBASE

**FILENAME="App\Data\Sequencing.avi" NOT ALLOWED**

SYSGUARDFILE="App\osguard.cp"

HREF="rtsp://App-V:554/app.sft"

GUID="A2A6D95E-8898-41C0-8020-797B23A1E917"

SIZE="33363442"/>

**Example 6 – Pre-Launching a Control Panel Applet within the Sequence**

<DEPENDENCY>

<SCRIPT TIMING="PRE" EVENT="LAUNCH" WAIT="TRUE" PROTECT="TRUE">

**<SCRIPTBODY>%SFT\_MNT%\\App\\CPL\\SYSDM.CPL</SCRIPTBODY>**

</SCRIPT>

</DEPENDENCY>

**Example 6a – Pre-Launching a Control Panel Applet within the Sequence**

<DEPENDENCY>

<SCRIPT TIMING="PRE" EVENT="LAUNCH" WAIT="TRUE" PROTECT="TRUE">

<SCRIPTBODY>

**"C:\\Program Files\\Common Files\\System\\Mapi\\1033\\mlcfg32.cpl"**

</SCRIPTBODY>

</SCRIPT>

</DEPENDENCY>

## <HREF> Examples

**Example 1 – Accessing Local Application**

<DEPENDENCY>

<SCRIPT TIMING="PRE" EVENT="LAUNCH" WAIT="TRUE" PROTECT="TRUE">

**<HREF>c:\winnt\system32\cmd.exe</HREF>**

</SCRIPT>

</DEPENDENCY>

**Example 1a – Accessing Batch File via UNC**

<DEPENDENCY>

<SCRIPT TIMING="PRE" EVENT="LAUNCH" WAIT="TRUE" PROTECT="TRUE">

**<HREF>\\SRV\_NAME\NT\_SHR\PRE\_SCRIPT.CMD</HREF>**

</SCRIPT>

</DEPENDENCY>

**NOTE:** HREF tag cannot be used with %SFT\_MNT% or to launch EXE from within virtual environment.

## <PARAMETERS> Examples

The PARAMETERS attribute is used to provide command line arguments to the FILENAME attribute. The FILENAME attribute points to the location of the program to execute when launching the virtual application. An example is placing filename in the PARAMETERS attribute, like a database file that will be opened when the application is launched.

**NOTE:** The PARAMETERS tag will not support environment variables (e.g. %SFT\_MNT%).

**Example 1 – Generic Example**

<CODEBASE

**FILENAME="\ABCD\EXECUTE\Z\_RUN.EXE"**

**PARAMETERS="FORM=MRF01 LANG=ENG ZENV=X\_POINT\_CHAR"**

SYSGUARDFILE="ACBS\osguard.cp"

HREF="rtsp://hercules:554/ACBS\_V2.sft"

GUID="BD9AF94E-4A8E-415D-B57F-54A454911FAA"

SIZE="648019522"/>

**Example 2 – Launching Data File via a Relative Path**

<CODEBASE

**FILENAME="App\WordEditor\wordpad.exe"**

**PARAMETERS=" ..\Data\cli.doc"**

SYSGUARDFILE="App\osguard.cp"

HREF="rtsp://App-V:554/app.sft"

GUID="A2A6D95E-8898-41C0-8020-797B23A1E917"

SIZE="33363442"/>

**NOTE:** ‘..\Data\cli.doc’ is located in the following location within the sequence:

%SFT\_MNT%\App\Data\cli.doc

## <WORKINGDIR> Examples – Specifying a Working Directory

The WORKINGDIR tag defines where to look for filenames that are not given a full path as in Windows. In the first example below, it is used make the C:\temp directory the default location to look for files if a referenced file name has no path.

**Example 1**

<IMPLEMENTATION>

**<WORKINGDIR>C:\temp</WORKINGDIR>**

<CODEBASE HREF="rtsp://ds1-soft1:554/np\_ecj.sft?Customer=Sequencer"

FILENAME="np\_ecj\notepad.exe" PARAMETERS="cjm.txt"

GUID="D47C0A23-32CF-11D6-990A-00A0CC6413D5"/>

<OS VALUE="WinXP"/>

</IMPLENENTATION>

**Example 2**

<IMPLEMENTATION>

**<WORKINGDIR>C:\program files\internet explorer</WORKINGDIR>**

<CODEBASE HREF="rtsp://ds1-soft1:554/np\_ecj.sft?Customer=Sequencer"

FILENAME="np\_ecj\notepad.exe" PARAMETERS="cjm.txt"

GUID="D47C0A23-32CF-11D6-990A-00A0CC6413D5"/>

<OS VALUE="WinXP"/>

</IMPLENENTATION>

**Example 3 – Example with Variable**

<IMPLEMENTATION>

**<WORKINGDIR>%SFT\_MNT%\App\Data</WORKINGDIR>**

<CODEBASE HREF="rtsp://ds1-soft1:554/np\_ecj.sft?Customer=Sequencer"

FILENAME="np\_ecj\notepad.exe" PARAMETERS="cjm.txt"

GUID="D47C0A23-32CF-11D6-990A-00A0CC6413D5"/>

<OS VALUE="WinXP"/>

</IMPLENENTATION>

## <REGISTRY> Examples – Registry Customizations

<IMPLEMENTATION>

<CODEBASE … />

<VIRTUALENV>

**<REGISTRY>**

**<REGKEY HIVE="HKLM" KEY="Software\Test">**

**<REGVALUE REGTYPE="REG\_SZ" NAME="string">Holas</REGVALUE>**

**<REGVALUE REGTYPE="REG\_DWORD" NAME="dword">5051</REGVALUE>**

**<REGVALUE REGTYPE="REG\_BINARY" NAME="hex">50,51,52</REGVALUE>**

**<REGVALUE REGTYPE="REG\_SZ">Value of Default Key</REGVALUE>**

**</REGKEY>**

**</REGISTRY>**

</VIRTUALENV>

</IMPLENENTATION>

## Launching Local Applications

Using the FILENAME attribute, the OSD file for an application does not have to open an executable within the virtual environment. In the case of Web add-ins like Microsoft Silverlight, the FILENAME attribute would be pointed to Internet Explorer. There are two examples listed below of launching local applications.

**NOTE:** The PARAMETERS tag will not support environment variables (e.g. %SFT\_MNT%).

**Example 1 – Launching Application that is local (not part of the Sequence)**

<CODEBASE

**FILENAME="C:\Program Files\Internet Explorer\iexplore.exe"**

SYSGUARDFILE="App\osguard.cp"

HREF="rtsp://App-V:554/app.sft"

GUID="A2A6D95E-8898-41C0-8020-797B23A1E917"

SIZE="33363442"/>

<VM VALUE="Win32">

<SUBSYSTEM VALUE=”Windows”/>

</VM>

<CODEBASE

**FILENAME=" C:\Program Files\Windows Media Player\wmplayer.exe"**

**PARAMETERS=" q:\app\data\sequencing.avi"**

SYSGUARDFILE="App\osguard.cp"

HREF="rtsp://App-V:554/app.sft"

GUID="A2A6D95E-8898-41C0-8020-797B23A1E917"

SIZE="33363442"/>

<VM VALUE="Win32">

<SUBSYSTEM VALUE=”Windows”/>

</VM>

**NOTE:** C:\Program Files\Windows Media Player\wmplayer.exe is a local application.

## Setting Environment Variables

Using these steps, an environment variable can be set which could be used by the application in the virtual environment. In this example, the system PATH variable is appended by the Mount path (for eg. Q:) and Visual Studio path. Hence, the application would use this updated PATH environment variable when required.

<VIRTUALENV>

<ENVLIST>

<ENVIRONMENT VARIABLE="PATH">%PATH%;%SFT\_MNT%\vs2005\Microsoft SDK\bin\.;</ENVIRONMENT>

</ENVLIST>

</VIRTUALENV>

## Launching Application via Non-Default Provider Policy

During installation of the App-V Infrastructure, a Default Provider Policy is created and assigned usage permissions. Provider policies can be created and modified to affect the authentication, access permissions of virtual applications, and licensing. Multiple provider policies can be created so that one is used for applications where no license enforcement is needed and another where license enforcement is enabled. In this scenario, applications that need to use a provider policy that is not the default must have the OSD file modified. The following describes the change to the OSD file for a non-default provider policy (in bold), where Sales is the name of the non-default provider policy.

<CODEBASE

FILENAME="Photoshp\Photoshop.exe"

SYSGUARDFILE="Photoshp\osguard.cp"

HREF="rtsp://appvserver:554/adobephotoshop70.sft**?Customer=Sales**"

GUID="FE72DDCD-6E80-4ABE-8E82-8208A134CB34"

SIZE="178299260"/>

## Running an Application as a CONSOLE application

THE SUBSYSTEM tag defines the type of application defined in the FILENAME attribute that will be run with two possible elements (**windows or console)** to define the type of application (windows=windows and console=dos) being executed. In this example, the OSD is describing that the application is a 16-bit DOS application. The values for the VMVALUE tag are Win16, Win32, and Win64 and are case sensitive.

<OS VALUE="WinXP"/>

<OS VALUE="Win2003TS"/>

<OS VALUE="WinVista"/>

<OS VALUE="Win2008TS"/>

<OS VALUE="Win7"/>

<OS VALUE="WinXP64"/>

<OS VALUE="Win2003TS64"/>

<OS VALUE="WinVista64"/>

<OS VALUE="Win2008TS64"/>

<OS VALUE="Win764"/>

<OS VALUE="Win2008R2TS64"/>

<VIRTUALENV>

<ENVLIST/>

</VIRTUALENV>

**<VM VALUE="Win16">** Can also be “Win32” or “Win64”􀃅 **WARNING! VM Values are case sensitive.**

**<SUBSYSTEM VALUE="console"/>**

**</VM>**

</IMPLEMENTATION>

## OSD XML Syntax Check

To check the XML format of an OSD file, change the .OSD file extension to .XML and drop the .XML file into an Internet Explorer window. If there are any open or missing tags the file will not be displayed.

Note: XML file editors such as XML Notepad can also be used to check XML file format.

Example XML file syntax display: <?xml version="1.0" standalone="no" ?>

**-** <SOFTPKG GUID="**26998703-328D-41DE-A256-5AFACD27ED08**" NAME="**WinZip**"

VERSION="**90**">

**-** <IMPLEMENTATION>

<CODEBASE

HREF="**rtsp://ntlabsg01:554/Winzip\_Winzip\_901\_MNT/Winzip\_Winzip\_901\_**

**MNT.sft**" GUID="**A9800C2A-E9ED-4711-B90C-808D6DD0AABC**"

FILENAME="**Winzip.v1\WINZIP32.EXE**" PARAMETERS=""

SYSGUARDFILE="**Winzip.v1\osguard.cp**" SIZE="**6225393**" />

<OS VALUE="**WinXP**" />

<OS VALUE="**WinVista**" />

<OS VALUE="**Win7**" />

<OS VALUE="**Win2003Svr**" />

<OS VALUE="**Win2003TS**" />

<OS VALUE="**Win2003TS64**" />

<OS VALUE="**Win2008TS**" />

<OS VALUE="**Win2008TS64**" />

<OS VALUE="**Win2008R2TS64**" />

<OS VALUE="**WinXP64**" />

<OS VALUE="**WinVista64**" />

<OS VALUE="**Win764**" />

**-** <VM VALUE="**Win32**">

<SUBSYSTEM VALUE="**windows**" />

</VM>

**-** <VIRTUALENV>

<ENVLIST />

</VIRTUALENV>

</IMPLEMENTATION>

**-** <DEPENDENCY>

<CLIENTVERSION VERSION="**2.0.5.0**" />

</DEPENDENCY>

<SUITE NAME="**Winzip\_Winzip\_901\_MNT**" />

<TITLE />

<ABSTRACT>**Win XP Sp1, IE 6.0 Seq Build 300**</ABSTRACT>

**-** <MGMT\_SHORTCUTLIST>

<SHORTCUT LOCATION="**%CSIDL\_STARTMENU%**" WORKINGDIR=""

DISPLAY="**WinZip**" ICON="**%SFT\_MIME\_SOURCE%/WinZip90.ico**" />

<SHORTCUT LOCATION="**%CSIDL\_DESKTOPDIRECTORY%**" WORKINGDIR=""

DISPLAY="**WinZip**" ICON="**%SFT\_MIME\_SOURCE%/WinZip90.ico**" />

</MGMT\_SHORTCUTLIST>

</SOFTPKG>

# Advanced Sequencing Techniques

This section will briefly cover some sequencing challenges and how to overcome them. It is also recommended to be aware of the logs created when sequencing, a list of the log files related to the Sequencer process and purpose.

## Sequencing Applications That Cannot Install to Q:\

At some point while sequencing applications, you will most likely run across an application that will not let you install to Q:\ (PowerPoint Viewer). This by itself does not present a problem when sequencing the application. In fact, only one minor modification will need to be made.

* Allow the application installer to install to its default location.

There are pros and cons when sequencing with this method.

## Sequencing Web-Based Applications

Internet Explorer is NOT supported in a sequenced application. However, you can sequence plug-ins or ActiveX controls for Internet Explorer.

Figure 30: Sequencing Web-Based Applications Process

This pulls the locally installed application into the virtual environment with the additions you have specified. In this case it was a Web plug-in. This lets you have a clean and secure Internet Explorer that you can lock down on a user’s machine and then allow them to use the plug-ins that you define.

## Sequencing Applications that Require Access to Local Devices and/or Resources

In many ways, you sequence these applications no different than you would any other. The applications inside the virtual environment can see the local resources on the OS and can access them. Devices are another matter. App-V cannot virtualize drivers. If you can install the driver separately from the application, then the application will fall through to the OS to look for the driver and see that it is installed.

## Sequencer Log Files

* ***Sft-seq-log.txt***  
  Description: The main log file for logging on the Sequencer.  
  Path: ProgramFiles%\Microsoft Application Virtualization Sequencer\Logs\Sft-seq-log.txt
* ***Sftrbt.txt***  
  Description: Stores the actions taken during the reboot processing.  
  Path: %ProgramFiles%\Microsoft Application Virtualization Sequencer \App-V Sequencer\Logs\Sftrbt.txt
* ***SFTCallBack.txt***  
  Description: Records all process starts and stops seen by SystemGuard.  
  Path: %ProgramFiles%\Microsoft Application Virtualization Sequencer \App-V Sequencer\Logs\SFTCallBack.txt
* ***Filter.log***  
  Description: A binary log that stores file activity during sequencing.  
  Path: %ProgramFiles%\Microsoft Application Virtualization Sequencer \App-V Sequencer\Logs\Filter.1.log
* ***Files.txt***  
  Description: Logs all files in the VFS. It must be enabled by setting.   
  HKLM\Software\Softricity\App-V  
  Sequencer\CurrentVersion\Configuration\FileManifest to “1”. (REG\_DWORD)  
  Path: %ProgramFiles%\Microsoft Application Virtualization Sequencer\App-V Sequencer\Logs\File.txt
* ***Devirtualizer\_log.txt***

Description: Records activity related to unpacking App-V packages when using the Expand to Local System function.

Path: %ProgramFiles%\Microsoft Application Virtualization Sequencer \App-V Sequencer\Logs\Devirtualizer\_log.txt

# Finding Additional Information

## Error Codes and the Microsoft Knowledge Base

Microsoft has completed the process of migrating data off of the old Softricity knowledge base. The 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 12 digits are unique to the version of the App-V Client software. The ending 8 digits of the error code remain constant between all client versions.

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

e.g., xxxxxx - xxxxxxxx - xxxxxxxx

or

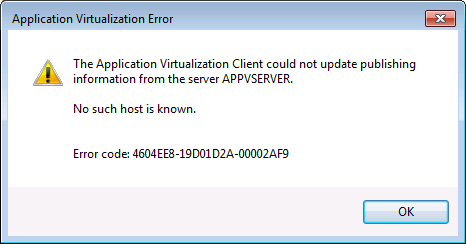


Figure 31: Sample Error Message

The first block of 6 characters are around the version build of the platform/client. Basically, when you see these, it’s information-based and will vary depending on:

* Major Version
* Minor Version
* Patch Version
* Build

The Next Block of 8 Characters Identify

* File ID
* Line
* Info

The Last Block of 8 Characters show:

* Problem Code

The last 10 digits of the error code are constant between the App-V builds. Because the last 10 digits are consistent we tend to use these for identifying and searching out errors. Regardless of the version, builds, etc., we can use this knowledge to identify problems and issues as these codes are reused between modules.

In addition, Justin Zarb from Premiere Field Engineering in the UK has written an excellent article related to the App-V error codes and what they mean. The article can be found at:

<http://blogs.technet.com/virtualworld/archive/2007/07/27/so-what-do-all-these-errors-actually-mean.aspx>

## App-V Team Blog

This 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 Related 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.brianmadden.com>
* <http://blogs.technet.com/virtualworld>
* <http://social.technet.microsoft.com/Forums/en-US/category/appvirtualization>