FLEXERA SOFTWARE

FLEXERA SOFTWARE® AdminStudio®

AdminStudio 2016 SP2

Evaluation Guide

Legal Information

Book Name:	AdminStudio 2016 SP2 Evaluation Guide
Part Number:	ADS-2016SP2-EG00
Product Release Date:	01 August 2017

Copyright Notice

Intellectual Property

For a list of trademarks and patents that are owned by Flexera Software, see http://www.flexerasoftware.com/intellectual-property. All other brand and product names mentioned in Flexera Software products, product documentation, and marketing materials are the trademarks and registered trademarks of their respective owners.

Restricted Rights Legend

The Software is commercial computer software. If the user or licensee of the Software is an agency, department, or other entity of the United States Government, the use, duplication, reproduction, release, modification, disclosure, or transfer of the Software, or any related documentation of any kind, including technical data and manuals, is restricted by a license agreement or by the terms of this Agreement in accordance with Federal Acquisition Regulation 12.212 for civilian purposes and Defense Federal Acquisition Regulation Supplement 227.7202 for military purposes. The Software was developed fully at private expense. All other use is prohibited.

Contents

AdminStudio 2016 SP2 Evaluation Guide

AdminStudio Suite powers an enterprise's daily Application Readiness process for packaging, and compatibility testing of physical, virtual, and mobile applications, ensuring faster service delivery and predictable deployment.

AdminStudio's automation, integration with leading software distribution tools, and workflow capabilities help companies streamline day-to-day application packaging operations and reduce the time and cost of application virtualization, Windows migrations, and mobile application deployment.

Trusted by more than 10,000 organizations, AdminStudio Suite eliminates the need for multiple standalone tools and provides an integrated solution that is adaptive and scalable to changing business priorities. AdminStudio provides enterprise IT teams the most advanced solution to efficiently prepare reliable application packages for deployment with a complete suite of automated MSI packaging, customization, testing, and reporting capabilities.

You can use this Evaluation Guide to quickly learn how to use AdminStudio to prepare Windows Installer, virtual applications, web applications, macOS desktop applications, and mobile applications for deployment. This Evaluation Guide is organized into the following sections:

Table 1 • AdminStudio 2016 SP2 Evaluation Guide

Section	Description
Getting Started	Explains what is included in this Evaluation Guide and how to get started.
New Features in AdminStudio 2016 SP2	Provides an overview of the new features in AdminStudio 2016 SP2.
New Features in AdminStudio 2016 SP1	Provides an overview of the new features in AdminStudio 2016 SP1.
New Features in AdminStudio 2016	Provides an overview of the new features in AdminStudio 2016.

Table 1 • AdminStudio 2016 SP2 Evaluation Guide

Section	Description
Setting Up AdminStudio Infrastructure	Explains how to perform the one-time set-up tasks that you need to complete prior to using AdminStudio including creating an Application Catalog, configuring a virtual machine, configuring e-mail settings, and specifying server connection settings.
Migrating to Windows Installer	Explains how to convert a sample setup to a Windows Installer package, import it into the Application Catalog, test it for operating system compatibility, best practices, and conflicts, and then prepare it for distribution.
Migrating to Application Virtualization	Explains how to use the Automated Application Converter to convert Windows Installer packages to virtual packages, and then test and distribute the virtual packages.
Testing for Application Compatibility	Explains how to use Application Manager to test packages for compatibility with Microsoft Windows 7, Windows 8, Windows 10, Windows Server 2008 R2, and Windows Server 2012 platforms, as well as to test web applications for compatibility with Internet Explorer 9/10/11 and Microsoft Edge.

Getting Started

This section explains what is included in this Evaluation Guide and how to get started:

- About Evaluating AdminStudio
- Purpose of Evaluation Guide
- Organization of Evaluation Guide
- Evaluation Guide Data Files

About Evaluating AdminStudio

You can choose to evaluate AdminStudio for 21 days. By clicking **Continue to Evaluate AdminStudio** dialog box that opens when you launch AdminStudio, you can begin evaluating the AdminStudio Enterprise Edition client tools.

Information about evaluating the AdminStudio client tools includes the following topics:

- AdminStudio Client Tools Evaluation Restrictions
- Evaluating AdminStudio's Microsoft App-V Support
- Evaluating the Automated Application Converter "Multiple Application" Option

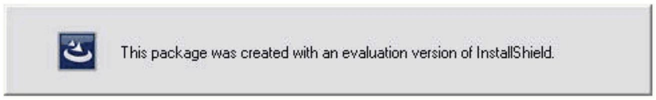
AdminStudio Client Tools Evaluation Restrictions

When you run AdminStudio in trial/evaluation mode, all of its features are fully available, with the following restrictions:

- **Can create only one Application Catalog**—You are permitted to create only one Application Catalog, and it must be named `AdminStudio Evaluation Catalog`.
- **Ten package import limit**—Only 10 total packages (of one or more deployment types) can be imported into the Application Catalog.
- **Package deletion not permitted**—After you import a package into the Application Catalog, you are not permitted to delete it.
- **AdminStudio Platform API support is disabled**—All platform support is disabled.

Evaluating AdminStudio's Microsoft App-V Support

While evaluating the AdminStudio Enterprise Edition client tools, you will be able to convert a Windows Installer package to an App-V application using the Automated Application Converter, Conversion Wizard, Repackager, and the InstallShield App-V Assistant. However, an App-V application built using an evaluation version of AdminStudio will display the following message every time it is launched:



This package was created with an evaluation version of InstallShield.

Figure 1: Evaluation Version Message

After purchasing the AdminStudio Virtualization Pack, you will be able to remove this message by rebuilding the App-V application.

Evaluating the Automated Application Converter "Multiple Application" Option

The Multiple Application option of Automated Application Converter is only available when you purchase the Virtualization Pack with AdminStudio Enterprise Edition.

***Note** • If you purchase the Virtualization Pack with AdminStudio Standard or Professional Editions, you will only be able to convert one package at a time, using one virtual machine.*

When using an evaluation version of AdminStudio, you will be able to use Automated Application Converter to convert a directory full of Windows Installer packages into individual virtual packages, but the conversion will be limited to three packages per run, using only one virtual machine. Therefore, only the first three packages that Automated Application Converter encounters will be converted to virtual applications.

Purpose of Evaluation Guide

The purpose of this Evaluation Guide is to help system administrators and other reviewers learn how to quickly get started using AdminStudio to prepare Windows Installer and virtual applications for deployment. By performing the exercises in this Evaluation Guide using the provided data files, you will learn how to:

- Create an Application Catalog
- Repackage a sample package
- Import packages into the Application Catalog
- Test Windows Installer packages, App-V packages, and web applications
- Configure a virtual machine for use in automated repackaging
- Convert Windows Installer packages to virtual applications
- Distribute Windows Installer and App-V packages
- Perform operating system compatibility and browser compatibility testing

Organization of Evaluation Guide

This Evaluation Guide provides exercises that guide you through performing the following key procedures:

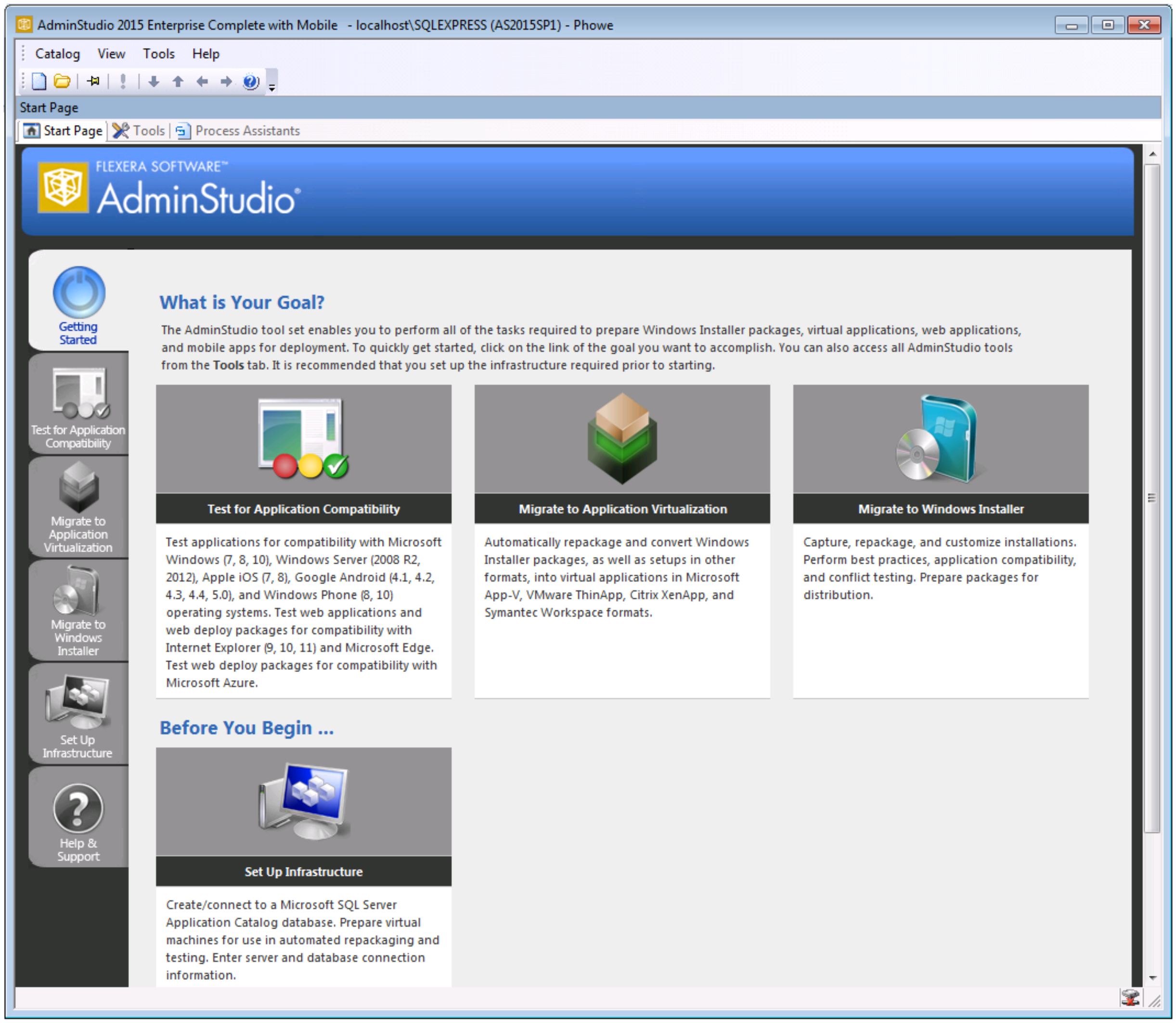
Table 2 • Evaluation Exercises

Procedure	Procedure
Setting Up AdminStudio Infrastructure	In these set of exercises, you will perform the one-time setup tasks that are necessary to get started using AdminStudio: ● Creating an Application Catalog ● Configuring a Virtual Machine ● Setting E-Mail Notification Settings ● Entering Server/Database Connection Settings
Migrating to Windows Installer	In this set of exercises, you will migrate a sample setup (such as an `.exe` file) to a deployable Windows Installer package (`.msi`): ● Repackaging a Sample Package ● Importing Packages into the Application Catalog ● Testing a Repackaged Application and Resolving Issues ● Distributing a Repackaged Application

Table 2 • Evaluation Exercises

Procedure		Procedure
	Migrating to Application Virtualization	In this set of exercises, you will migrate a set of applications into virtual applications that are ready for deployment: ● Identifying Packages to Virtualize ● Converting to Virtual Formats ● Testing and Distributing Converted Packages
	Testing for Application Compatibility	In this set of exercises, you will test Windows Installer packages for application readiness on Microsoft Windows (7, 8, 10), and Windows Server (2008 R2 and 2012). You will also test web applications for compatibility with Internet Explorer 9/10/11 and Microsoft Edge. ● Importing Packages, Web Applications, and Mobile Apps ● Selecting Tests to Run and Setting Default Fix Option ● Performing Testing and Viewing Results

These four main procedures are also featured on AdminStudio’s Start Page.


AdminStudio 2015 Enterprise Complete with Mobile - localhost\SQLEXPRESS (AS2015SP1) - Phowe
Catalog
View
Tools
Help
Start Page
Start Page
Tools
Process Assistants
FLEXERA SOFTWARE™
AdminStudio®
Getting Started
Test for Application Compatibility
Migrate to Application Virtualization
Migrate to Windows Installer
Set Up Infrastructure
Help & Support

What is Your Goal?

The AdminStudio tool set enables you to perform all of the tasks required to prepare Windows Installer packages, virtual applications, web applications, and mobile apps for deployment. To quickly get started, click on the link of the goal you want to accomplish. You can also access all AdminStudio tools from the **Tools** tab. It is recommended that you set up the infrastructure required prior to starting.

Test for Application Compatibility

Test applications for compatibility with Microsoft Windows (7, 8, 10), Windows Server (2008 R2, 2012), Apple iOS (7, 8), Google Android (4.1, 4.2, 4.3, 4.4, 5.0), and Windows Phone (8, 10) operating systems. Test web applications and web deploy packages for compatibility with Internet Explorer (9, 10, 11) and Microsoft Edge. Test web deploy packages for compatibility with Microsoft Azure.

Migrate to Application Virtualization

Automatically repackage and convert Windows Installer packages, as well as setups in other formats, into virtual applications in Microsoft App-V, VMware ThinApp, Citrix XenApp, and Symantec Workspace formats.

Migrate to Windows Installer

Capture, repackage, and customize installations. Perform best practices, application compatibility, and conflict testing. Prepare packages for distribution.

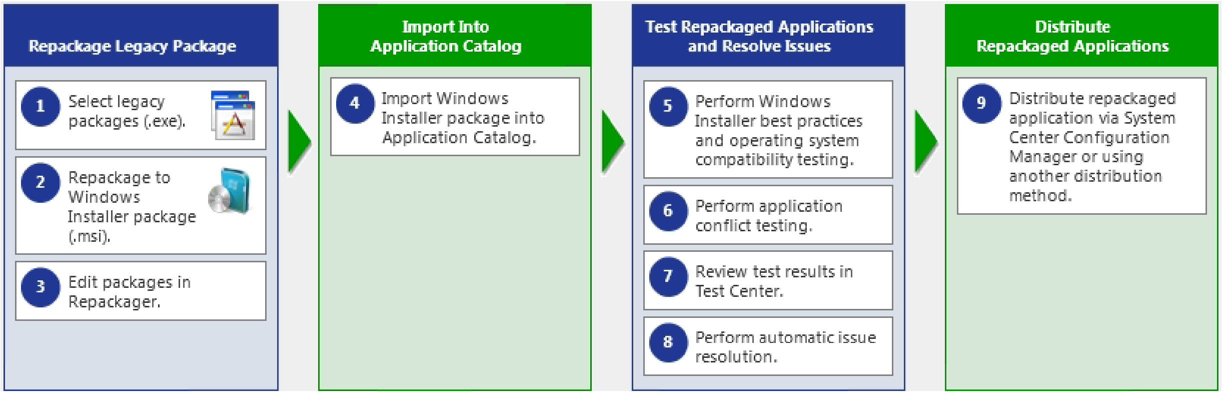
Before You Begin ...

Set Up Infrastructure

Create/connect to a Microsoft SQL Server Application Catalog database. Prepare virtual machines for use in automated repackaging and testing. Enter server and database connection information.

Figure 2: AdminStudio Start Page

Each subtab of the Start Page includes a flowchart that lists the steps in each procedure. For example, the following flowchart is displayed on the **Migrate to Windows Installer** tab:



Repackage Legacy Package

1 Select legacy packages (.exe).

2 Repackage to Windows Installer package (.msi).

3 Edit packages in Repackager.

Import Into Application Catalog

4 Import Windows Installer package into Application Catalog.

Test Repackaged Applications and Resolve Issues

5 Perform Windows Installer best practices and operating system compatibility testing.

6 Perform application conflict testing.

7 Review test results in Test Center.

8 Perform automatic issue resolution.

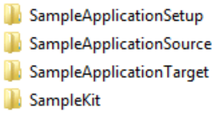
Distribute Repackaged Applications

9 Distribute repackaged application via System Center Configuration Manager or using another distribution method.

Figure 3: Migrate to Windows Installer Tab of AdminStudio Start Page

Evaluation Guide Data Files

To perform the exercises in the AdminStudio Evaluation Guide, you will be using the sample data that is provided in the `AS2016EvalGuideDataFiles.zip` file. These data files are organized into the following directories:



SampleApplicationSetup
SampleApplicationSource
SampleApplicationTarget
SampleKit

Figure 4: Directory Structure of AdminStudio Evaluation Guide Data Files

These data files demonstrate the recommended organizational structure that you should use when you want to import a directory of packages into the Application Catalog:

- **One root directory**—Organize the packages you want to import in one root directory (`AS2016EvalGuideDataFiles` in this example).
- **Each application in a subdirectory**—Each application should be stored in its own first level subdirectory (such as `SampleKit` or `SampleApplicationTarget`).
- **Each deployment type in a sub-subdirectory**—Each deployment type should be stored in its own sub-subdirectory (`AppV`, `MSI`, etc.) of the application directory.

Unzip this data file and place its contents in a location accessible to your installation of AdminStudio, such as:

```
C:\AS2016EvalGuideDataFiles
```

New Features in AdminStudio 2016 SP2

This section lists the new features that are included in AdminStudio 2016 SP2:

- Support for PowerShell Wrapped Packages
- Support for Additional Types of Complex Installer Executables
- Java Detection Tests and Java Dependency Reports for Desktop Applications
- Best Practices and Risk Assessment Tests Split into Two Categories
- New Windows Server 2016 Compatibility Tests in Test Center
- Support for System Center Configuration Manager (Current Branch)
- Ability to Add Destination Group During System Center Configuration Manager Distribution
- Ability to Select Connection on Microsoft Configuration Manager Deployments Report
- Ability to Set Temporary Directory Location
- InstallShield 2016 SP2

Support for PowerShell Wrapped Packages

AdminStudio 2016 SP2 provides full support for PowerShell-wrapped packages, including converting Windows Installer packages (`.msi`) and complex installation packages (`.exe`) to PowerShell wrapped packages (PowerShell script `.ps1` files) using the PowerShell App Deployment Toolkit.

PowerShell is a task automation and configuration management framework from Microsoft. The PowerShell App Deployment Toolkit provides a set of functions to perform common application deployment tasks and to interact with the user during a deployment. By wrapping an existing Windows Installer or complex installation package in a PowerShell script, you are able to automatically perform deployment tasks both before and after installation.

AdminStudio 2016 SP2 also supports importing existing PowerShell-wrapped packages into the Application Catalog, testing a PowerShell-wrapped package's bundled child packages using Test Center, testing PowerShell-wrapped packages on a virtual machine, and distributing PowerShell wrapped packages using Distribution Wizard. You can also convert a PowerShell-wrapped package to a virtual package using Automated Application Converter or the Conversion Wizard.

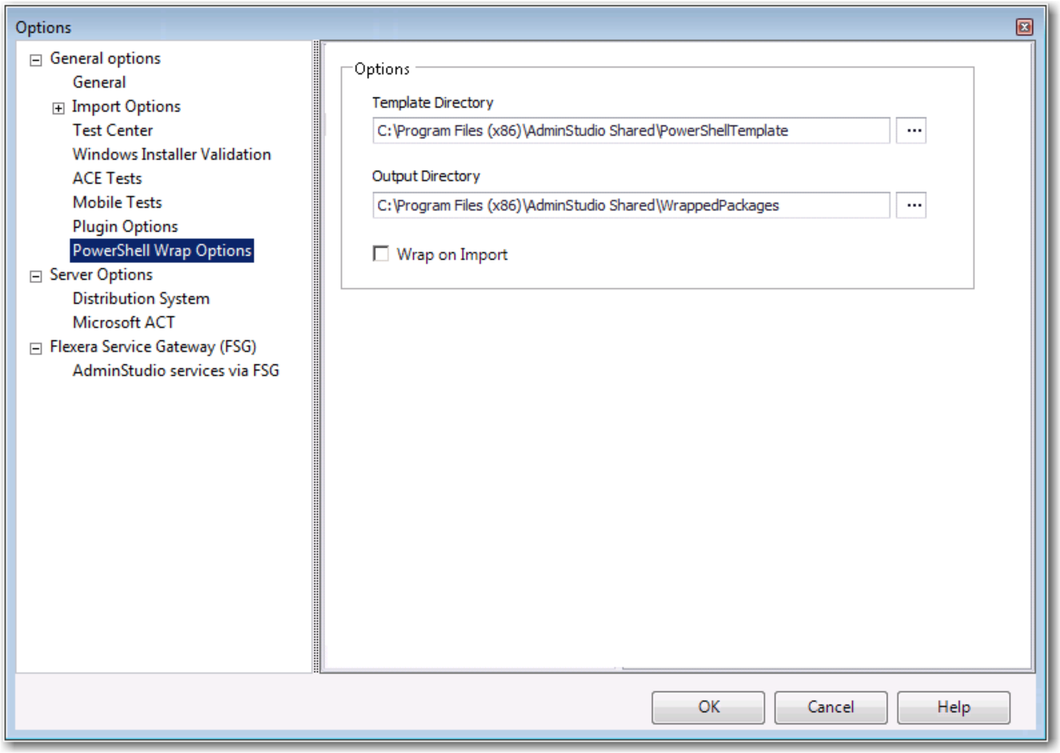
Information about AdminStudio 2016 SP2's support for PowerShell-wrapped packages is organized in the following topics:

- Setting PowerShell Wrap Options
- PowerShell Script Template File
- Wrapping a Windows Installer (.msi) or Installer Executable (.exe)
- Importing PowerShell Wrapped Packages
- Viewing a Power Shell Wrapped Package's Bundled Child Applications
- Testing PowerShell Wrapped Packages
- PowerShell Wrapped Packages Report

- Distributing PowerShell Wrapped Packages
- Virtualizing PowerShell Wrapped Packages
- Testing PowerShell Wrapped Packages on a Virtual Machine
- Customizing the PowerShell Installer
- Editing a PowerShell-Wrapped Package
- API Support for Performing PowerShell Wrapping: Invoke-ASPowerShellWrap

Setting PowerShell Wrap Options

On a new tab of the Application Manager **Options** dialog box, **PowerShell Wrap Options**, you can specify the location of the PowerShell template, the output directory for PowerShell wrapped packages, and set an option to automatically wrap a package during import.



Template Directory: C:\Program Files (x86)\AdminStudio Shared\PowerShellTemplate

Output Directory: C:\Program Files (x86)\AdminStudio Shared\WrappedPackages

□ Wrap on Import

Figure 5: PowerShell Wrap Options Tab of Options Dialog Box

The **PowerShell Wrap Options** tab includes the following properties:

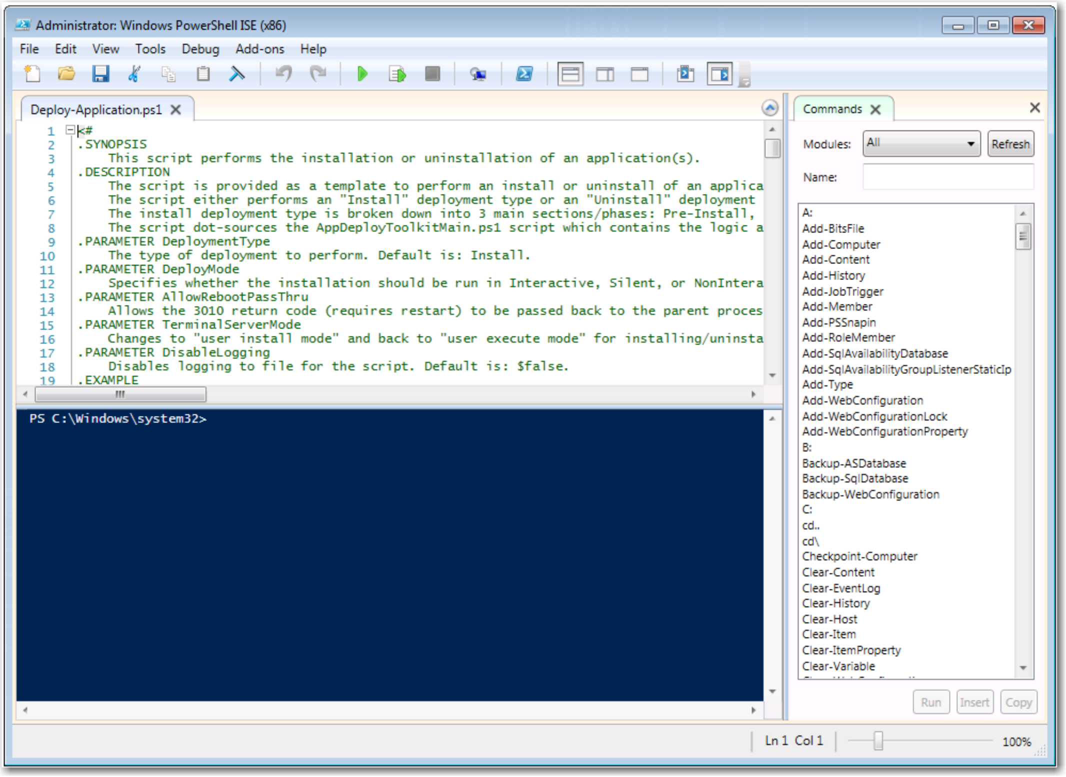
Option	Description
Template Directory	Specify the location of the template that you want to use to create PowerShell script files. A default template, `Deploy-Application.ps1`, along with other necessary files are provided in the following location: `C:\Program Files (x86)\AdminStudio Shared\PowerShellTemplate\` You can click the browse button and select a different directory. For more information, see PowerShell Script Template File.
Output Directory	Specify the location where the PowerShell-wrapped packages that you create using AdminStudio will be stored. By default, the location is: `C:\Program Files (x86)\AdminStudio Shared\WrappedPackages\` You can click the browse button and select a different directory.
Wrap on Import	Select this option if you want to automatically convert Windows Installer packages (`.msi`) and complex installation packages (`.exe`) to PowerShell wrapped packages (PowerShell script `.ps1` files) upon import into the Application Catalog. By default, this option is not selected.

PowerShell Script Template File

A PowerShell script template file, `Deploy-Application.ps1`, is provided in the following location:

```
C:\Program Files (x86)\AdminStudio Shared\PowerShellTemplate\
```

You can customize this script to meet your deployment requirements by editing it in the Windows PowerShell ISE application.



```
<#
.SYNOPSIS
    This script performs the installation or uninstallation of an application(s).
.DESCRIPTION
    The script is provided as a template to perform an install or uninstall of an applica
    The script either performs an "Install" deployment type or an "Uninstall" deployment
    The install deployment type is broken down into 3 main sections/phases: Pre-Install,
    The script dot-sources the AppDeployToolkitMain.ps1 script which contains the logic a
.PARAMETER DeploymentType
    The type of deployment to perform. Default is: Install.
.PARAMETER DeployMode
    Specifies whether the installation should be run in Interactive, Silent, or NonIntera
.PARAMETER AllowRebootPassThru
    Allows the 3010 return code (requires restart) to be passed back to the parent proces
.PARAMETER TerminalServerMode
    Changes to "user install mode" and back to "user execute mode" for installing/uninsta
.PARAMETER DisableLogging
    Disables logging to file for the script. Default is: $false.
.EXAMPLE
```

```
PS C:\Windows\system32>
```

Figure 6: PowerShell Script Template File, Deploy-Application.ps1, in Windows PowerShell ISE

This script is provided as a template to perform an install or uninstall of an application(s). The script either performs an “install” deployment type or an “uninstall” deployment type. The install deployment type is broken down into three main sections: Pre-Install, Install, and Post-Install. The script refers to the `AppDeployToolkitMain.ps1` script which contains the logic and functions required to install or uninstall an application.

Contents of the PowerShellTemplate Directory

In addition to the `Deploy-Application.ps1` template file, the `PowerShellTemplate` directory also includes these additional files:

File	Description
`Deploy-Application.exe` `Deploy-Application.exe.config`	Executable that invokes a PowerShell script.
`Deploy-Application.ps1`	PowerShell script template file which is modified by the PowerShell Package Wrap Wizard to contain the actual install logic for the specific package, including: ● Installer Path ● Name ● Version ● Manufacturer ● Language ● Install Command Line ● Uninstall Command Line
`AppDeployToolkit\AppDeployToolkitBanner.png`	Banner image which can be customized to brand the PowerShell installer.
`AppDeployToolkit\AppDeployToolkitLogo.ico`	Image used when the PowerShell installer runs.
`AppDeployToolkit\AppDeployToolkitConfig.xml`	Contains configurable preferences for how the toolkit behaves, including language settings.
`AppDeployToolkit\AppDeployToolkitExtensions.ps1`	Script containing your extensions for the framework.
`AppDeployToolkit\AppDeployToolkitMain.ps1`	Main script with all functions.
`AppDeployToolkit\AppDeployToolkitHelp.ps1`	Help file explaining the functions.
`AppDeployToolkit\AppDeployToolkitMain.cs`	Additional functions.

Rules for a Custom PowerShell Template

If you are creating a custom PowerShell template file, observe the following rules:

- **Do not rename Deploy-Application.ps1**—The `Deploy-Application.exe` application looks for a file named `Deploy-Application.ps1`. Therefore, if the `.ps1` file name is changed to the package name or anything else, the install/uninstallation will fail.
- **Deploy-Application.ps1 must be in the root folder**—The `Deploy-Application.exe` application expects the `Deploy-Application.ps1` file to be present in the root folder level, the same folder that contains the `Deploy-Application.exe` file.
- **Installer package must be in the Files folder**—The installer package (`.msi` or `.exe`) must be in the `Files` folder.
- **AppDeployToolkit must be in the root folder**—The `AppDeployToolkit`folder (along with all of its contents) must be in the root folder.

Note • *Installation/uninstallation is not obstructed if the `Deploy-Application.exe.config` file is not present in the parent folder level or is even deleted.*

Note • *The `$dirFiles` variable includes the path to the installer package. If you change the value of this variable, installation will fail.*

Wrapping a Windows Installer (.msi) or Installer Executable (.exe)

In AdminStudio 2016 SP2, you can convert a Windows Installer (`.msi`) or complex installation package (`.exe`) to a PowerShell wrapped package (PowerShell script `.ps1` file) by selecting it in the Application Manager tree and selecting **Wrap package**.

By wrapping an existing Windows Installer or complex installation package in a PowerShell script, you are able to launch additional deployment tasks via the PowerShell script both before and after installation. The advantages of using PowerShell wrapping instead of repackaging or transforms to modify an installer include:

- **Package is not modified**—The original package is not modified.
- **Consistent UI experience**—You can provide a consistent UI experience to your users for all of your organization’s installers.
- **Powerful PowerShell functionality**—The PowerShell scripting language provides a more powerful set of functionality than can be provided using transforms.

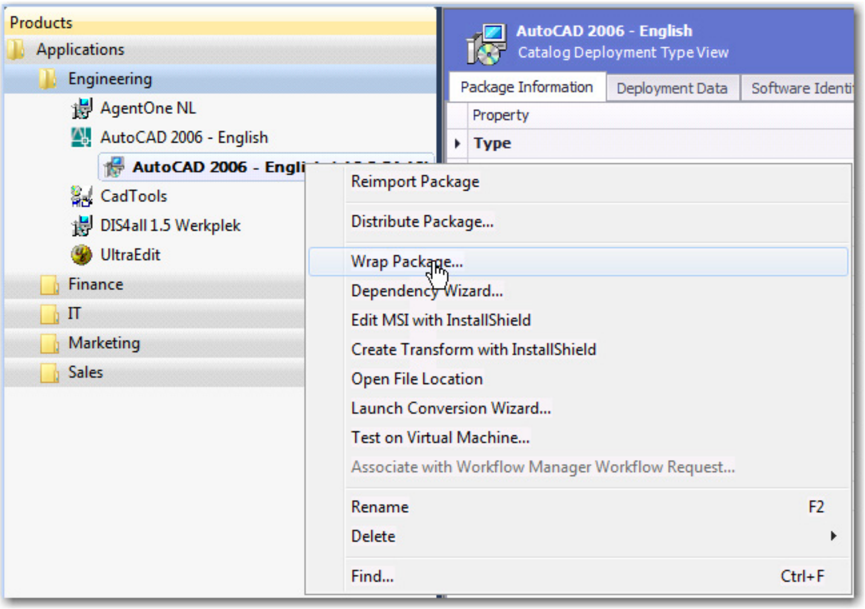
You can convert the following deployment types into a PowerShell wrapped package:

- Windows Installer packages (`.msi`)
- InstallShield InstallScript `.exe` files
- InstallShield Basic MSI installers that are compressed into a `setup.exe` file
- InstallShield Suite Installer `.exe` files
- Wise Package Studio `.exe` files
- Other executable file types that can be uncompressed by 7-ZIP

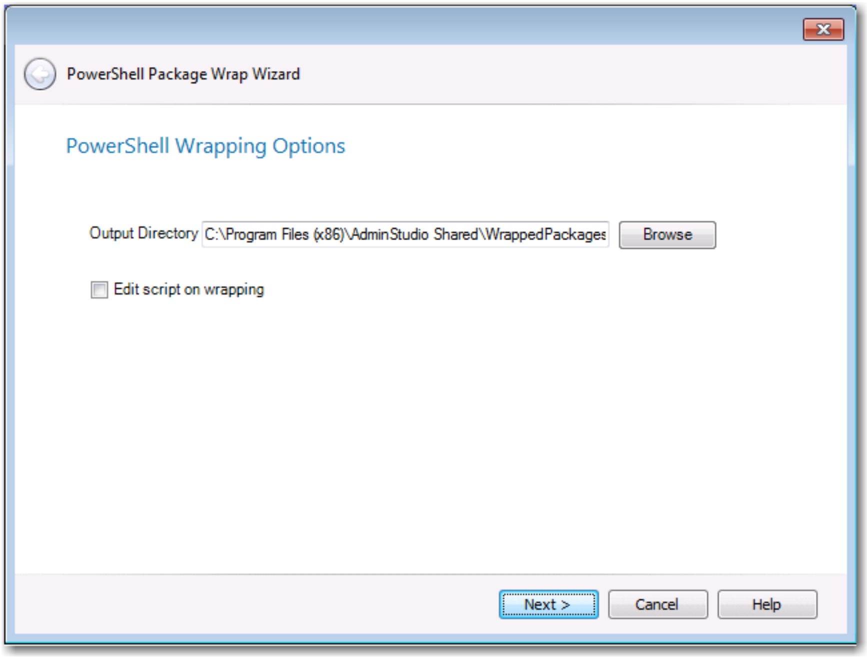
To convert a Windows Installer (`.msi`) or installation package (`.exe`) to a PowerShell-wrapped package, perform the following steps.

Task ***To convert a Windows Installer (.msi) or installation package (.exe) to a PowerShell wrapped package (.ps1):***

1. Select the Windows Installer (`.msi`) or installation package (`.exe`) package in the Application Manager tree and select **Wrap Package** from the context menu.



The **PowerShell Package Wrap Wizard** opens.



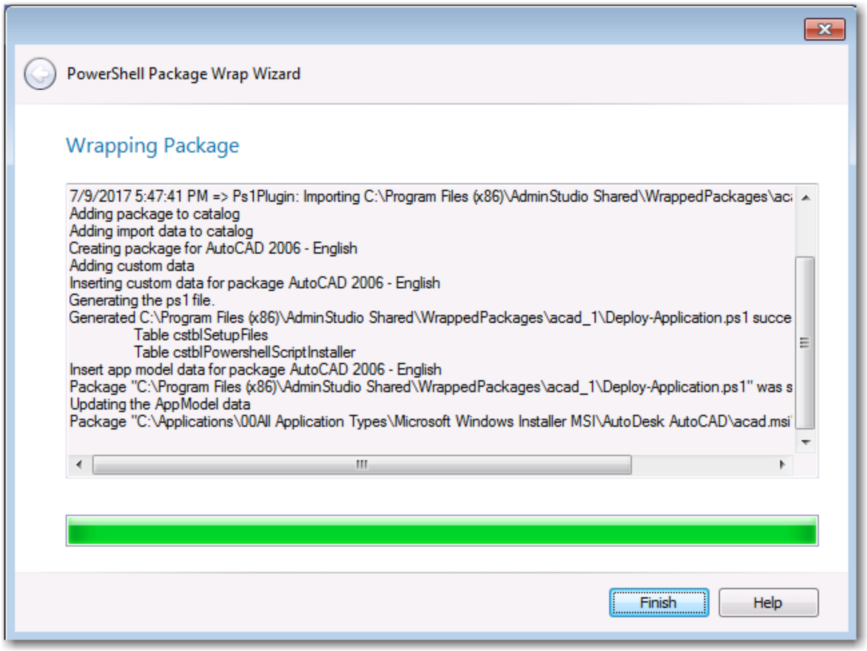
PowerShell Package Wrap Wizard

PowerShell Wrapping Options

Output Directory C:\Program Files (x86)\AdminStudio Shared\WrappedPackages Browse

□ Edit script on wrapping

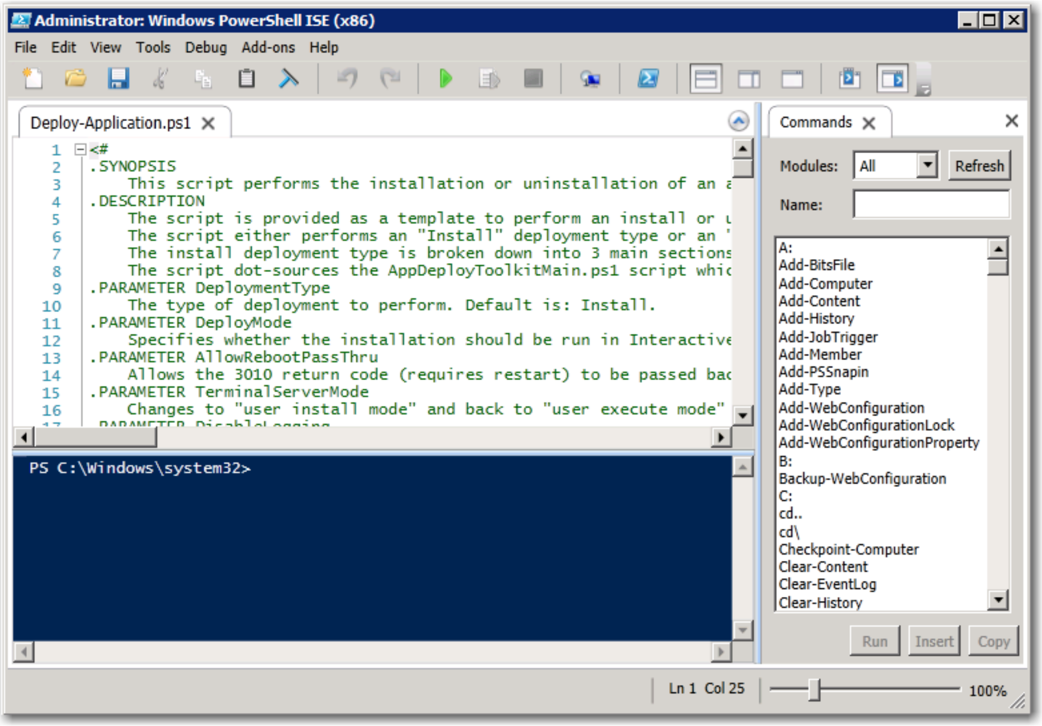
2. Specify the **Output Directory**, the location where the PowerShell-wrapped package will be stored.
3. If you want to edit the default PowerShell script file in Windows PowerShell ISE after the package has been wrapped, select **Edit script on wrapping** option.
4. Click **Next**. The **Wrapping Package** panel opens and wrapping is initiated. When the process is complete, confirmation messages are listed.



```
7/9/2017 5:47:41 PM => Ps1Plugin: Importing C:\Program Files (x86)\AdminStudio Shared\WrappedPackages\ac:
Adding package to catalog
Adding import data to catalog
Creating package for AutoCAD 2006 - English
Adding custom data
Inserting custom data for package AutoCAD 2006 - English
Generating the ps1 file.
Generated C:\Program Files (x86)\AdminStudio Shared\WrappedPackages\acad_1\Deploy-Application.ps1 succe
            Table cstblSetupFiles
            Table cstblPowershellScriptInstaller
Insert app model data for package AutoCAD 2006 - English
Package "C:\Program Files (x86)\AdminStudio Shared\WrappedPackages\acad_1\Deploy-Application.ps1" was s
Updating the AppModel data
Package "C:\Applications\00All Application Types\Microsoft Windows Installer MSI\AutoDesk AutoCAD\acad.msi
```

5. Click **Finish** to close the wizard.

If you selected the **Edit script on wrapping** option, the Windows PowerShell ISE application opens and you are prompted to edit the PowerShell script file that you just created.

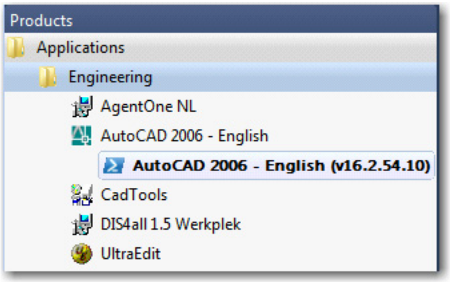


```
<#
.SYNOPSIS
    This script performs the installation or uninstallation of an a
.DESCRIPTION
    The script is provided as a template to perform an install or u
    The script either performs an "Install" deployment type or an '
    The install deployment type is broken down into 3 main sections
    The script dot-sources the AppDeployToolkitMain.ps1 script whic
.PARAMETER DeploymentType
    The type of deployment to perform. Default is: Install.
.PARAMETER DeployMode
    Specifies whether the installation should be run in Interactive
.PARAMETER AllowRebootPassThru
    Allows the 3010 return code (requires restart) to be passed bac
.PARAMETER TerminalServerMode
    Changes to "user install mode" and back to "user execute mode"
.PARAMETER DisableLogging
```

```
PS C:\Windows\system32>
```

6. Make any desired changes and save the script file.

7. In the Application Manager tree, view the package you just wrapped. You will see that the `.msi` or `.exe` deployment type icon for this package has been replaced with a PowerShell deployment type icon.

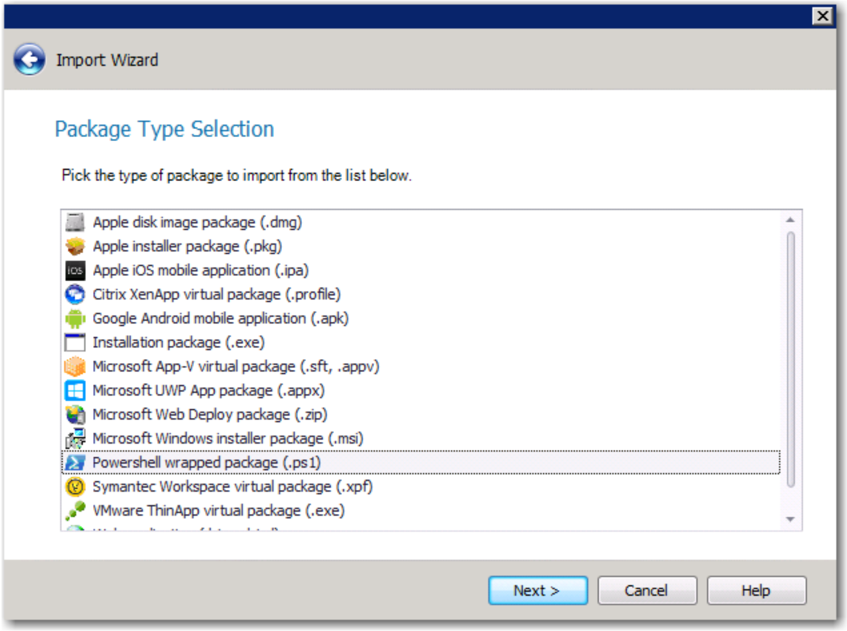


Important *• All of the OS compatibility, application virtualization compatibility and best practices ICE tests results that were generated by testing the package before it was wrapped are carried over and are viewable on* ***Test Center*** *tab when you select its new PowerShell deployment type icon.*

Note *• Log files for the PowerShell Package Wrap Wizard can be found at:* `C:\Windows\Logs\Software`*.*

Importing PowerShell Wrapped Packages

In AdminStudio 2016 SP2, you can import existing PowerShell wrapped packages into the Application Catalog. The **Package Type Selection** panel of the **Import Wizard** now has a new selection type: **PowerShell wrapped package (.ps1)**.



Package Type Selection

Pick the type of package to import from the list below.

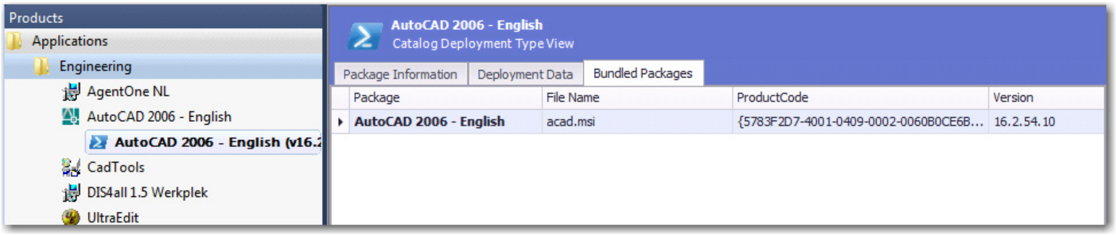
- Apple disk image package (.dmg)
- Apple installer package (.pkg)
- Apple iOS mobile application (.ipa)
- Citrix XenApp virtual package (.profile)
- Google Android mobile application (.apk)
- Installation package (.exe)
- Microsoft App-V virtual package (.sft, .appv)
- Microsoft UWP App package (.appx)
- Microsoft Web Deploy package (.zip)
- Microsoft Windows installer package (.msi)
- Powershell wrapped package (.ps1)
- Symantec Workspace virtual package (.xpf)
- VMware ThinApp virtual package (.exe)

Figure 7: PowerShell Wrapped Package (.ps1) Option on Package Type Selection Panel of Import Wizard

***Note •** Also, if the **Wrap on Import** option is selected on the **PowerShell Wrap Options** tab of the **Options** dialog box, Windows Installer packages (`.msi`) and installation packages (`.exe`) packages will automatically be wrapped (converted to PowerShell script files) during import.*

Viewing a Power Shell Wrapped Package's Bundled Child Applications

You can view the child packages that are bundled within a PowerShell script file on the **Bundled Packages** tab of the **Catalog Deployment Type** view.

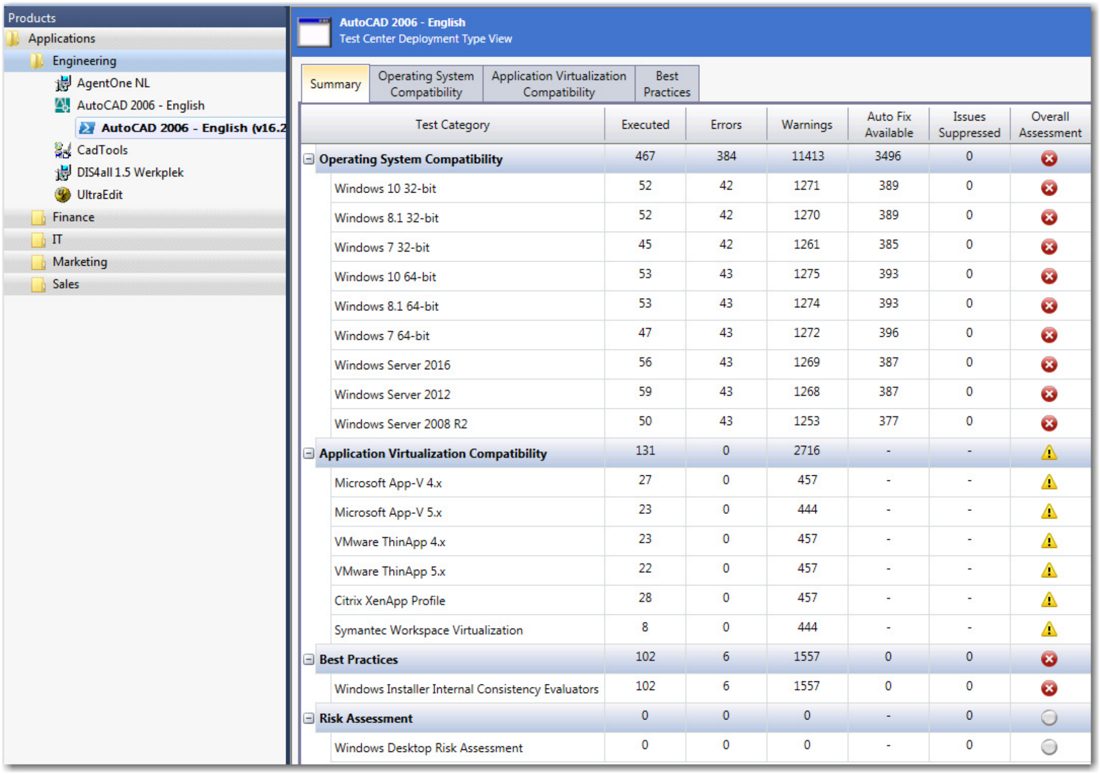


Package	File Name	ProductCode	Version
AutoCAD 2006 - English	acad.msi	{5783F2D7-4001-0409-0002-0060B0CE6B...	16.2.54.10

Figure 8: Displaying Child Packages of PowerShell Script File on Bundled Packages Tab

Testing PowerShell Wrapped Packages

Using Test Center, you can perform OS compatibility, application virtualization compatibility, and best practices testing of the child `.msi` and `.exe` packages that are bundled in a PowerShell script file.



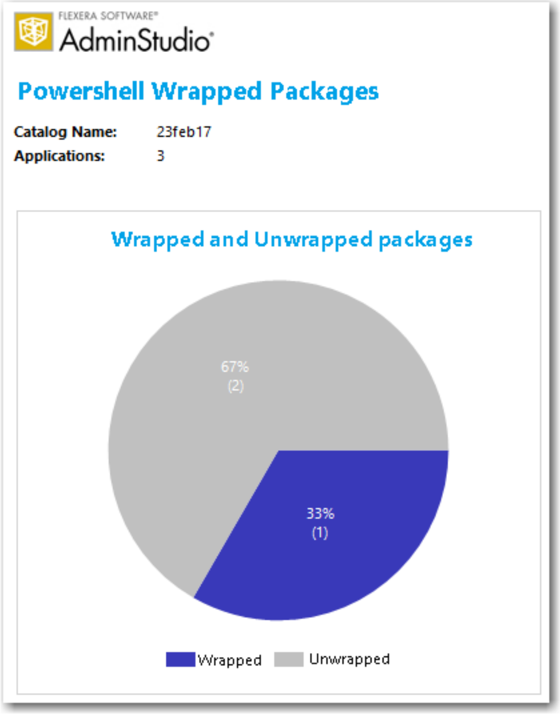
Test Category	Executed	Errors	Warnings	Auto Fix Available	Issues Suppressed	Overall Assessment
⊟ **Operating System Compatibility**	467	384	11413	3496	0	✖
Windows 10 32-bit	52	42	1271	389	0	✖
Windows 8.1 32-bit	52	42	1270	389	0	✖
Windows 7 32-bit	45	42	1261	385	0	✖
Windows 10 64-bit	53	43	1275	393	0	✖
Windows 8.1 64-bit	53	43	1274	393	0	✖
Windows 7 64-bit	47	43	1272	396	0	✖
Windows Server 2016	56	43	1269	387	0	✖
Windows Server 2012	59	43	1268	387	0	✖
Windows Server 2008 R2	50	43	1253	377	0	✖
⊟ **Application Virtualization Compatibility**	131	0	2716	-	-	⚠
Microsoft App-V 4.x	27	0	457	-	-	⚠
Microsoft App-V 5.x	23	0	444	-	-	⚠
VMware ThinApp 4.x	23	0	457	-	-	⚠
VMware ThinApp 5.x	22	0	457	-	-	⚠
Citrix XenApp Profile	28	0	457	-	-	⚠
Symantec Workspace Virtualization	8	0	444	-	-	⚠
⊟ **Best Practices**	102	6	1557	0	0	✖
Windows Installer Internal Consistency Evaluators	102	6	1557	0	0	✖
⊟ **Risk Assessment**	0	0	0	-	0	○
Windows Desktop Risk Assessment	0	0	0	-	0	○

Figure 9: Test Results for a PowerShell Wrapped Package

To test a PowerShell-wrapped package, open the **Test Center** tab, select the application or package in the Application Manager tree, and click **Execute Tests**.

PowerShell Wrapped Packages Report

In AdminStudio 2016 SP2, new tests have been added to Report Center for the PowerShell-wrapped packages deployment type.



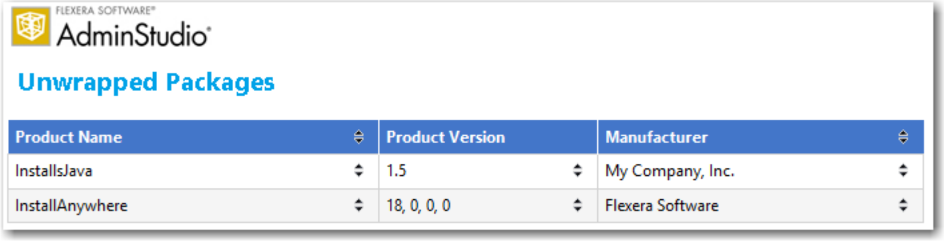
FLEXERA SOFTWARE® AdminStudio®

Powershell Wrapped Packages

Catalog Name: 23feb17
Applications: 3

Figure 10: PowerShell Wrapped Packages Summary Report

When you click on one of the sections of the pie chart, a detailed report opens listing packages in that category.



FLEXERA SOFTWARE® AdminStudio®

Unwrapped Packages

Product Name	Product Version	Manufacturer
InstallsJava	1.5	My Company, Inc.
InstallAnywhere	18, 0, 0, 0	Flexera Software

Figure 11: PowerShell Wrapped Package Detail Report

Distributing PowerShell Wrapped Packages

You can distribute PowerShell-wrapped packages to System Center Configuration Manager and Symantec Altiris Client Management Suite using the Distribution Wizard.

Select the application containing the PowerShell-wrapped package deployment type, and then click the **Distribute** button in the toolbar to open the Distribution Wizard, and proceed with distribution.

For more information, see Distributing Applications Using the Distribution Wizard.

Virtualizing PowerShell Wrapped Packages

In AdminStudio 2016 SP2, you can convert a PowerShell-wrapped package to a virtual package using the Conversion Wizard. For more information, see Using the Conversion Wizard.

Testing PowerShell Wrapped Packages on a Virtual Machine

You can test the installation of a PowerShell wrapped package on a virtual machine using the **Test on Virtual Machine Wizard**. This wizard uses the capability of the Automated Application Converter tool to spin up the selected virtual machine and install the selected package.

For more information, see Using Test on Virtual Machine Wizard.

Customizing the PowerShell Installer

You can customize the banner image that is displayed in the PowerShell installer and the post-installation messages that are displayed.

- Customizing the PowerShell Installer Banner Image
- Customizing Post-Installation Text Messages

Customizing the PowerShell Installer Banner Image

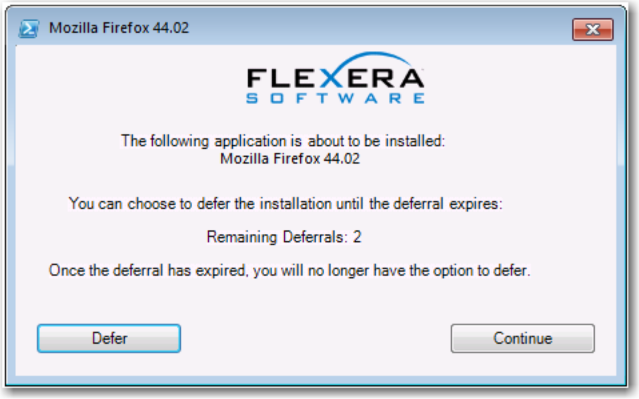
When AdminStudio converts a package to a PowerShell-wrapped package, the following image file is displayed at a banner at the top of the installer:

```
\AdminStudio Shared\PowerShellTemplate\AppDeployToolkit\AppDeployToolkitBanner.png
```

This image, by default, is the Flexera Software logo.

Figure 12: PowerShell Installer Default Banner

When this image is displayed in the PowerShell installer, it looks like this:


Mozilla Firefox 44.02

The following application is about to be installed:
Mozilla Firefox 44.02

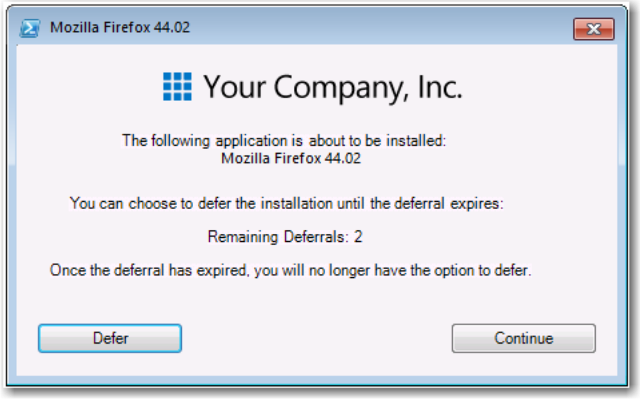
You can choose to defer the installation until the deferral expires:

Remaining Deferrals: 2

Once the deferral has expired, you will no longer have the option to defer.

Figure 13: Banner Displayed in PowerShell Installer

You can customize this banner by replacing the content of the `AppDeployToolkitBanner.png` file.



The following application is about to be installed:
Mozilla Firefox 44.02

You can choose to defer the installation until the deferral expires:

Remaining Deferrals: 2

Once the deferral has expired, you will no longer have the option to defer.

Figure 14: PowerShell Installer with Customized Banner

Customizing Post-Installation Text Messages

You can customize the post-installation messages that are displayed in the PowerShell installer by editing the `Show-InstallationPrompt -Message` value in the `Deploy-Application.ps1` file. By default, the message is: `Thank you for using Flexera Software!`.

```
##*===============================================
##* POST-INSTALLATION
##*===============================================
[string]$installPhase = 'Post-Installation'

## <Perform Post-Installation tasks here>

## Display a message at the end of the install
If (-not $useDefaultMsi) { Show-InstallationPrompt -Message 'Thank you for using Flexera Software!' -
ButtonRightText 'OK' -Icon Information -NoWait }
```

Editing a PowerShell-Wrapped Package

You can edit the PowerShell-wrapped package script (`.ps1` file) using the Windows PowerShell ISE application. If you selected the **Edit script on wrapping** option on the **PowerShell Wrapping Options** panel of the **PowerShell Package Wrap Wizard**, the script file opens in the Windows PowerShell ISE application immediately after you close the wizard.

You can also edit the `.ps1` file by right-clicking on the PowerShell-wrapped package node in the Application Manager tree, and then clicking **Edit Package** on the menu.

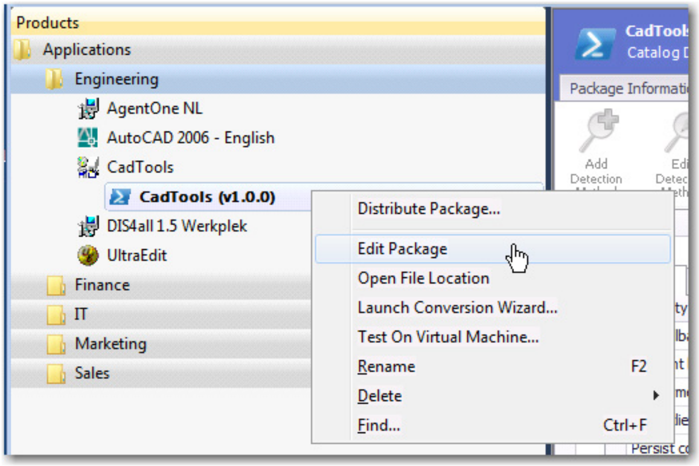


Figure 15: Editing a PowerShell-Wrapped Package

You can also launch the PowerShell editor by clicking the **Edit with PowerShell Editor** button in the ribbon.

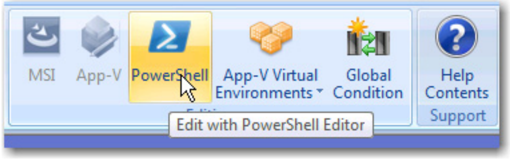
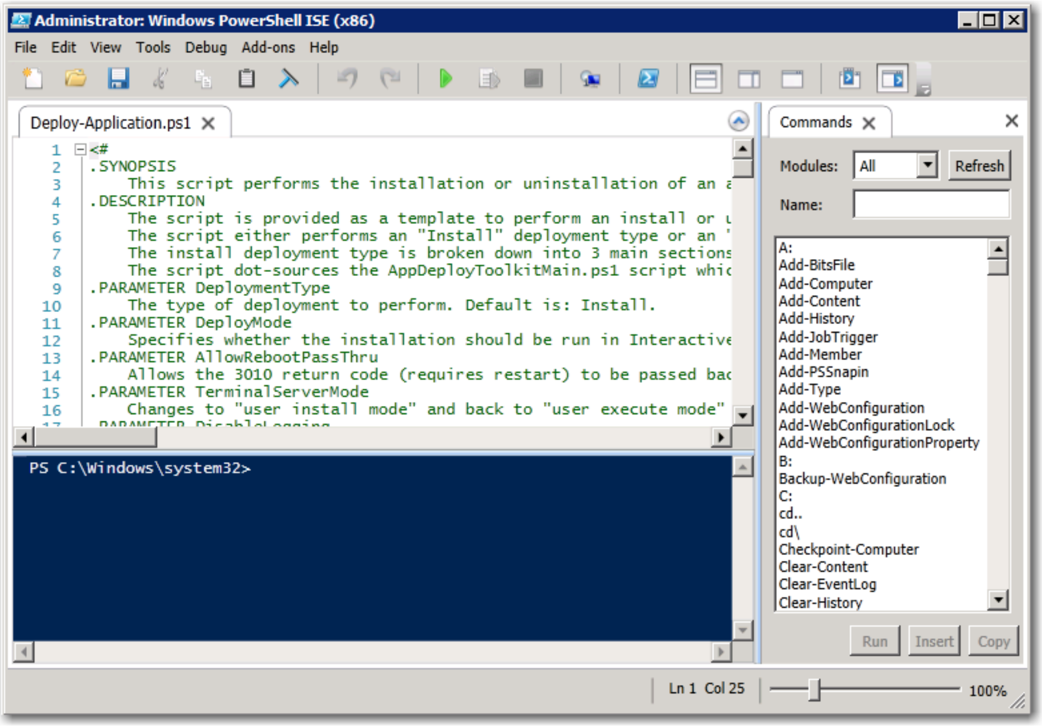


Figure 16: Edit with PowerShell Editor Button

Windows PowerShell ISE application opens, where you can edit the PowerShell script file.



```
<#
.SYNOPSIS
    This script performs the installation or uninstallation of an a
.DESCRIPTION
    The script is provided as a template to perform an install or u
    The script either performs an "Install" deployment type or an '
    The install deployment type is broken down into 3 main sections
    The script dot-sources the AppDeployToolkitMain.ps1 script whic
.PARAMETER DeploymentType
    The type of deployment to perform. Default is: Install.
.PARAMETER DeployMode
    Specifies whether the installation should be run in Interactive
.PARAMETER AllowRebootPassThru
    Allows the 3010 return code (requires restart) to be passed bac
.PARAMETER TerminalServerMode
    Changes to "user install mode" and back to "user execute mode"
.PARAMETER DisableLogging
```

```
PS C:\Windows\system32>
```

Figure 17: Editing a PowerShell Script File in Windows PowerShell ISE

API Support for Performing PowerShell Wrapping: Invoke-ASPowerShellWrap

In AdminStudio 2016 SP2, a new PowerShell API command has been added to AdminStudio to support performing PowerShell wrapping. You can use the `Invoke-ASPowerShellWrap` command to convert a Windows Installer (`.msi`) or installation package (`.exe`) to a PowerShell wrapped package file (`.ps1`).

The following is an example of the `Invoke-ASPowerShellWrap` command:

```
Invoke-ASPowerShellWrap -PackageId 25 -OutputDirectory "D\WrappedPackagesFeb"
```

For more information, see Invoke-ASPowerShellWrap.

Support for Additional Types of Complex Installer Executables

As described in Ability to Inspect and Test Complex Installer Executables, expanded support for complex installer executable files (which contain bundled Windows Installer packages) was introduced in AdminStudio 2016, allowing you to view the names of the bundled Windows Installer packages and to perform OS compatibility, application virtualization compatibility, and best practices testing on those bundled packages. At this time, support for the following complex installer executable types was provided:

- InstallShield InstallScript `.exe` files
- InstallShield Basic MSI installers that are compressed into a `setup.exe` file
- InstallShield Suite Installer `.exe` files
- Wise Package Studio `.exe` files
- Other executable file types that can be uncompressed by 7-ZIP

In AdminStudio 2016 SP2, enhanced identification of imported complex installer executables is provided. AdminStudio can now detect the following additional types and read information from those packages, including the following types:

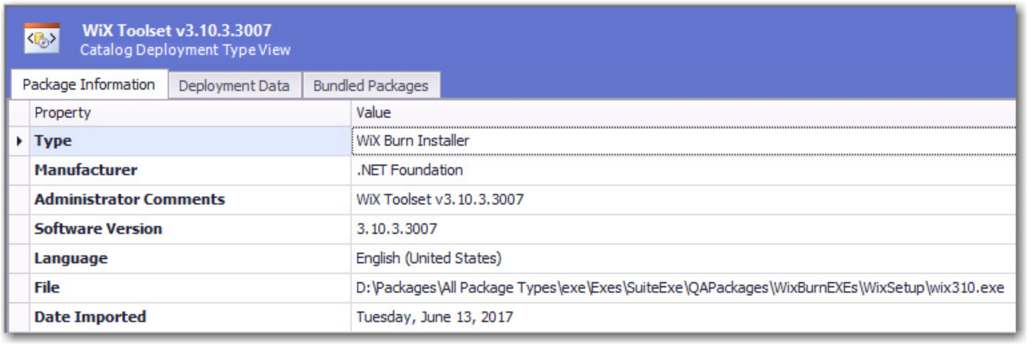
- InnoSetup

  ***Note** • See Enabling Support for InnoSetup Executables.*

- Nullsoft
- WiseScript
- WiX Burn
- Advanced Installer
- InstallAnywhere

***Note** • If AdminStudio is unable to identify the `.exe` type of an imported package, by default it is categorized as a legacy installer.*

These new legacy application types are identified in the **Type** field on the **Package Information** tab of the **Catalog Deployment Type View**, identified as a **WiX Burn Installer** in the following example:

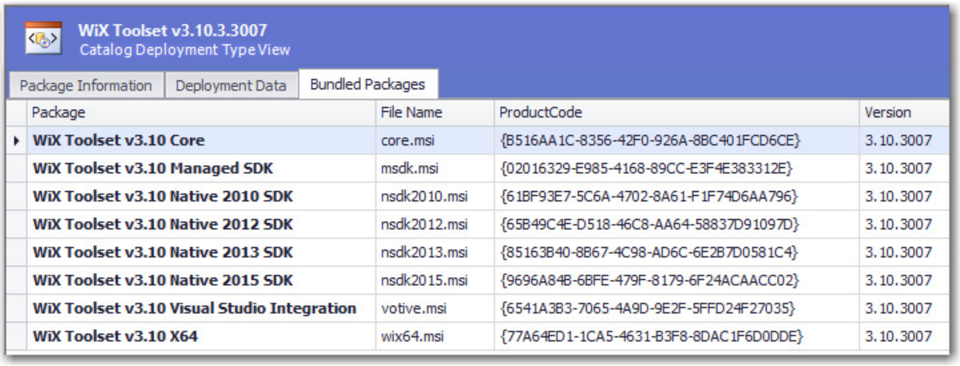


WiX Toolset v3.10.3.3007
Catalog Deployment Type View

Property	Value
Type	WiX Burn Installer
Manufacturer	.NET Foundation
Administrator Comments	WiX Toolset v3.10.3.3007
Software Version	3.10.3.3007
Language	English (United States)
File	D:\Packages\All Package Types\exe\Exes\SuiteExe\QAPackages\WixBurnEXEs\WixSetup\wix310.exe
Date Imported	Tuesday, June 13, 2017

Figure 18: Legacy Installer Type on Package Information Tab of Catalog Deployment Type View

You can also view the bundled packages of these newly supported legacy application types on the Bundled Packages tab of the Catalog Deployment Type view.

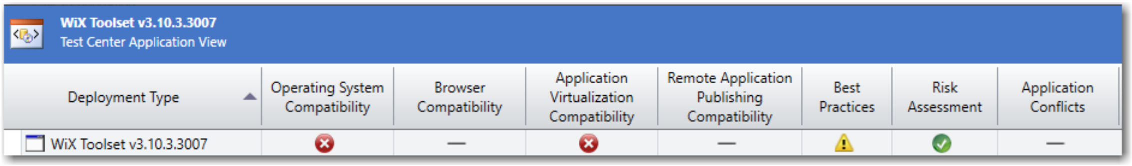


WiX Toolset v3.10.3.3007
Catalog Deployment Type View

Package	File Name	ProductCode	Version
WiX Toolset v3.10 Core	core.msi	{B516AA1C-8356-42F0-926A-8BC401FCD6CE}	3.10.3007
WiX Toolset v3.10 Managed SDK	msdk.msi	{02016329-E985-4168-89CC-E3F4E383312E}	3.10.3007
WiX Toolset v3.10 Native 2010 SDK	nsdk2010.msi	{61BF93E7-5C6A-4702-8A61-F1F74D6AA796}	3.10.3007
WiX Toolset v3.10 Native 2012 SDK	nsdk2012.msi	{65B49C4E-D518-46C8-AA64-58837D91097D}	3.10.3007
WiX Toolset v3.10 Native 2013 SDK	nsdk2013.msi	{85163B40-8B67-4C98-AD6C-6E2B7D0581C4}	3.10.3007
WiX Toolset v3.10 Native 2015 SDK	nsdk2015.msi	{9696A84B-6BFE-479F-8179-6F24ACAACC02}	3.10.3007
WiX Toolset v3.10 Visual Studio Integration	votive.msi	{6541A3B3-7065-4A9D-9E2F-5FFD24F27035}	3.10.3007
WiX Toolset v3.10 X64	wix64.msi	{77A64ED1-1CA5-4631-B3F8-8DAC1F6D0DDE}	3.10.3007

Figure 19: Bundled Packages of Newly Supported Complex Executable Installer Types

You can also test those bundled packages of the newly supported complete executable types in Test Center.



WiX Toolset v3.10.3.3007
Test Center Application View

Deployment Type	Operating System Compatibility	Browser Compatibility	Application Virtualization Compatibility	Remote Application Publishing Compatibility	Best Practices	Risk Assessment	Application Conflicts
WiX Toolset v3.10.3.3007	✖	—	✖	—	⚠	✓	—

Figure 20: Test Results of Bundled Packages of Newly Supported Complex Executable Installer Types

Enabling Support for InnoSetup Executables

AdminStudio 2016 SP2 supports the latest version of InnoSetup: `innounp.exe` (version 0.46.0.0).

In order for AdminStudio to extract child packages from InnoSetup executables, you need to perform the following steps.

Task ***To enable support for InnoSetup executables:***

1. Download the **InnoSetup Unpacker** RAR file (`innounp046.rar`) from the following location:

   `https://sourceforge.net/projects/innounp/files/innounp/innounp%200.46/innounp046.rar/download`

2. Uncompress the `innounp046.rar` file to obtain the `innounp.exe file`.

3. Copy the `innounp.exe` file to the following location on your AdminStudio machine:

   ```
   C:\Program Files (x86)\AdminStudio\2016\Common\Tools
   ```

Using Invoke-ASUpdatePackage to Extract Details on Bundled Child Packages of Imported Packages

Starting with AdminStudio 2016, when you import suite executables (`.exe`) that contain child packages, the child packages are automatically extracted during import and are available for testing.

However, if you are upgrading an existing Application Catalog (from releases prior to AdminStudio 2016) that already contains installers with bundled child packages, you will need to run the `Invoke-ASUpdatePackage` command on each of those existing installers to extract details on the child packages bundled with those installers.

The `Invoke-ASUpdatePackage` command is run on an existing `.exe` package to extract the bundled `.msi` details. Along with child package extraction, the `Invoke-ASUpdatePackage` also extracts Java dependency information and identifies the type of suite `.exe`.

The following is an example of the `Invoke-ASUpdatePackage` command:

```
Invoke-ASUpdatePackage -PackageId 2
```

For more information, see Invoke-ASUpdatePackage.

Java Detection Tests and Java Dependency Reports for Desktop Applications

AdminStudio 2016 SP2 includes new Java version dependency detection tests for desktop applications and Java dependency reports.

- Java Detection Tests
- Java Dependency Information
- Java Runtime Environment Dependency Report

Java Detection Tests

AdminStudio 2016 SP2 includes new Java detection tests for Windows and macOS desktop applications that determine whether an application has a dependency on a Java version (required during either package installation or after package installation) that has been designated as end of life/end of support. Either an Error or Warning is generated:

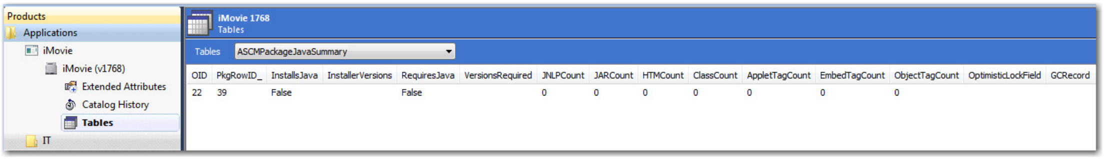
- **Error**—Generated if application is dependent upon a Java version that has been designated as end-of-life/end-of-support.
- **Warning**—Generated if application depends upon a version of Java that is not the most recent version.

The new tests are:

- EOLMAC001 and EOLWIN001: Java Versions Required
- EOLMAC002 and EOLWIN002: Java Installer Versions

Java Dependency Information

Java dependency information is extracted during package import of Windows and macOS desktop applications. You can view this Java dependency information for a package in a new table (`ASCMPackageJavaSummary`) viewable on the **Tables View**.



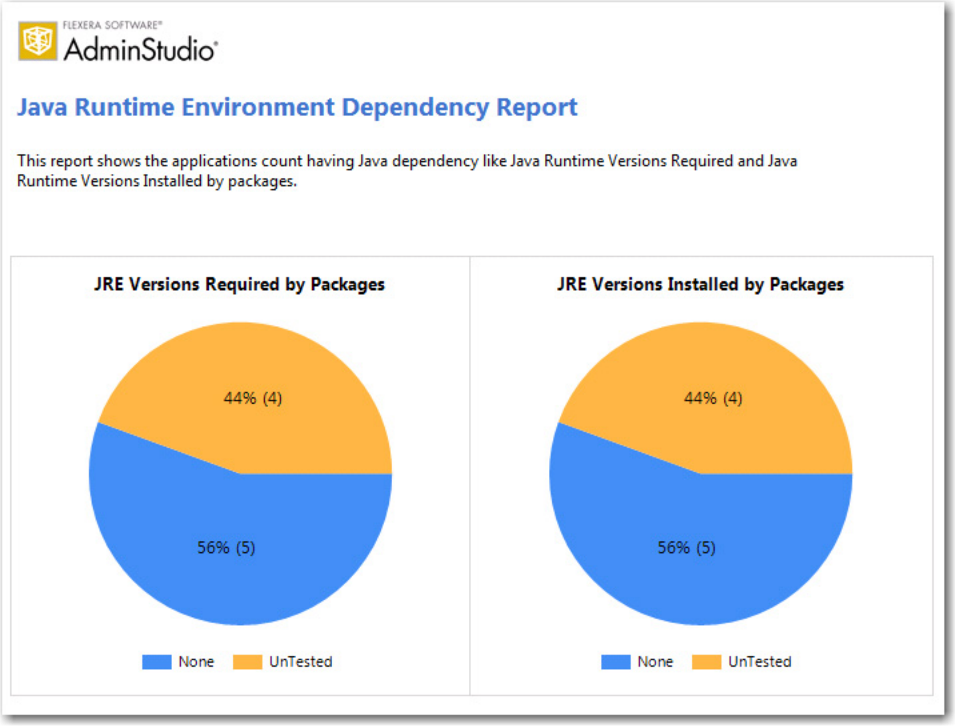
OID	PkgRowID_	InstallsJava	InstallerVersions	RequiresJava	VersionsRequired	JNLPCount	JARCount	HTMCount	ClassCount	AppletTagCount	EmbedTagCount	ObjectTagCount	OptimisticLockField	GCRecord
22	39	False		False		0	0	0	0	0	0	0		

Figure 21: ASCMPackageJavaSummary Table

Java Runtime Environment Dependency Report

A new Java Runtime Environment Dependency report has been introduced in AdminStudio 2016 SP2, which is opened by selecting **Dependencies > Java Runtime Dependencies** on the **Report Center** tab.

The initial view of this report shows the number of applications that require a Java Runtime version to run, and the number of applications that install a specific Java Runtime version during installation.



Java Runtime Environment Dependency Report

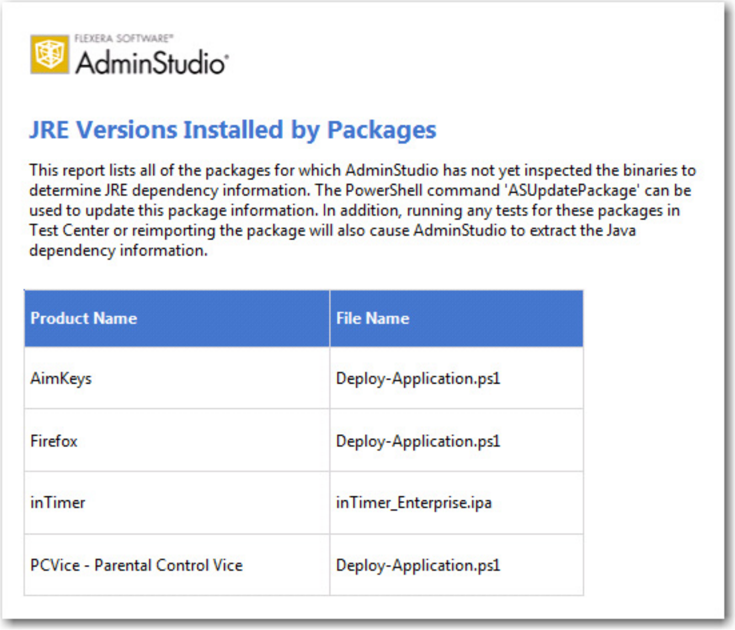
This report shows the applications count having Java dependency like Java Runtime Versions Required and Java Runtime Versions Installed by packages.

JRE Versions Required by Packages
44% (4)
56% (5)
None
UnTested
JRE Versions Installed by Packages
44% (4)
56% (5)
None
UnTested

Figure 22: Java Runtime Environment Dependency Report / Overview

When you click on these charts, you see more detailed information about the applications that fall into each category.

For example, the following is the detailed report on applications that install a Java Runtime version during installation:



FLEXERA SOFTWARE AdminStudio

JRE Versions Installed by Packages

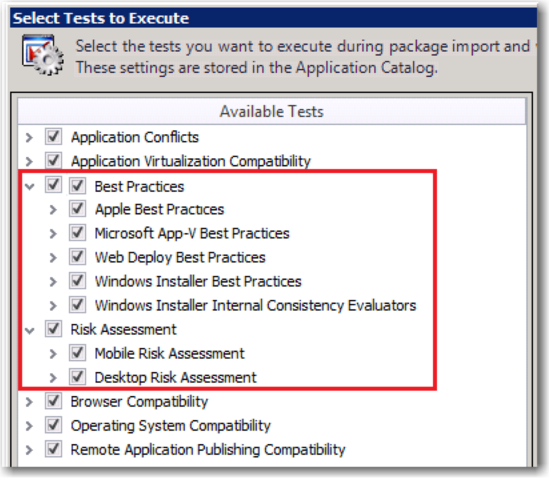
This report lists all of the packages for which AdminStudio has not yet inspected the binaries to determine JRE dependency information. The PowerShell command 'ASUpdatePackage' can be used to update this package information. In addition, running any tests for these packages in Test Center or reimporting the package will also cause AdminStudio to extract the Java dependency information.

Product Name	File Name
AimKeys	Deploy-Application.ps1
Firefox	Deploy-Application.ps1
inTimer	inTimer_Enterprise.ipa
PCVice - Parental Control Vice	Deploy-Application.ps1

Figure 23: JRE Versions Installed by Packages

Best Practices and Risk Assessment Tests Split into Two Categories

In AdminStudio 2016 SP2, the Test Center test group named **Best Practices and Risk Assessment Tests** has been split into two separate categories.



Select Tests to Execute

Select the tests you want to execute during package import and
These settings are stored in the Application Catalog.

Available Tests

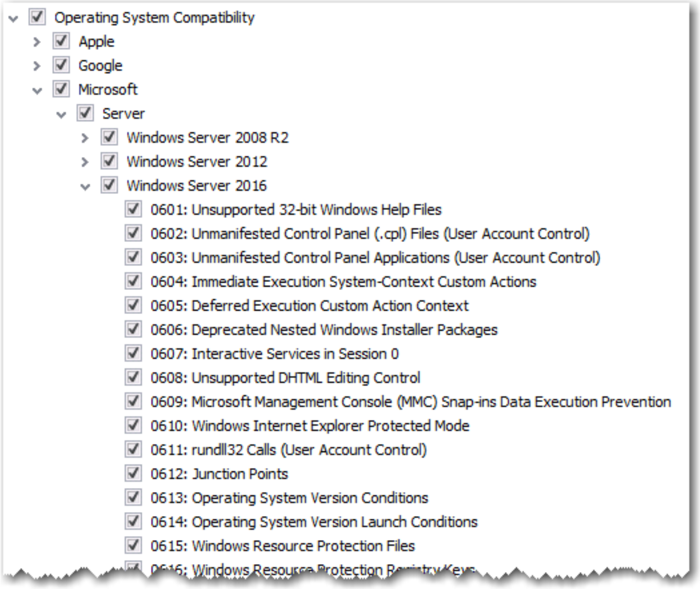
- ☑ Application Conflicts
- ☑ Application Virtualization Compatibility
- ☑ ☑ Best Practices
  - ☑ Apple Best Practices
  - ☑ Microsoft App-V Best Practices
  - ☑ Web Deploy Best Practices
  - ☑ Windows Installer Best Practices
  - ☑ Windows Installer Internal Consistency Evaluators
- ☑ Risk Assessment
  - ☑ Mobile Risk Assessment
  - ☑ Desktop Risk Assessment
- ☑ Browser Compatibility
- ☑ Operating System Compatibility
- ☑ Remote Application Publishing Compatibility

Figure 24: Best Practices and Rest Assessment Test Groups on the Select Tests to Execute Dialog Box

In previous releases, all of these tests were grouped in one category.

New Windows Server 2016 Compatibility Tests in Test Center

In AdminStudio 2016 SP2, 53 operating system compatibility test for Windows Server 2016 have been added.



- ☑ Operating System Compatibility
  - ☑ Apple
  - ☑ Google
  - ☑ Microsoft
    - ☑ Server
      - ☑ Windows Server 2008 R2
      - ☑ Windows Server 2012
      - ☑ Windows Server 2016
        - ☑ 0601: Unsupported 32-bit Windows Help Files
        - ☑ 0602: Unmanifested Control Panel (.cpl) Files (User Account Control)
        - ☑ 0603: Unmanifested Control Panel Applications (User Account Control)
        - ☑ 0604: Immediate Execution System-Context Custom Actions
        - ☑ 0605: Deferred Execution Custom Action Context
        - ☑ 0606: Deprecated Nested Windows Installer Packages
        - ☑ 0607: Interactive Services in Session 0
        - ☑ 0608: Unsupported DHTML Editing Control
        - ☑ 0609: Microsoft Management Console (MMC) Snap-ins Data Execution Prevention
        - ☑ 0610: Windows Internet Explorer Protected Mode
        - ☑ 0611: rundll32 Calls (User Account Control)
        - ☑ 0612: Junction Points
        - ☑ 0613: Operating System Version Conditions
        - ☑ 0614: Operating System Version Launch Conditions
        - ☑ 0615: Windows Resource Protection Files
        - ☑ 0616: Windows Resource Protection Registry Keys

Figure 25: Windows Server 2016 OS Compatibility Tests

Support for System Center Configuration Manager (Current Branch)

AdminStudio 2016 SP2 supports System Center Configuration Manager (Current Branch). You can connect Application Manager to a System Center Configuration Manager (Current Branch) server, which enables you to import packages from and distribute packages to that server.

***Note** • AdminStudio 2016 SP2 support for System Center Configuration Manager (Current Branch) was tested using Build 1610.*

Note the following about AdminStudio's support for System Center 2016 Configuration Manager:

- System Center Configuration Manager (Current Branch) only supports the **Fast Retry** type return code for Microsoft UWP app packages (`.appx`). The following return code types are not supported: **Success (no reboot)**, **Failure (no reboot)**, **Hard Reboot**, and **Soft Reboot**.

  ***Note** • UWP is the packaging format used to distribute and install apps on Windows 8.x and 10, and is the only format allowed for Universal Windows Platform (UWP) apps.*

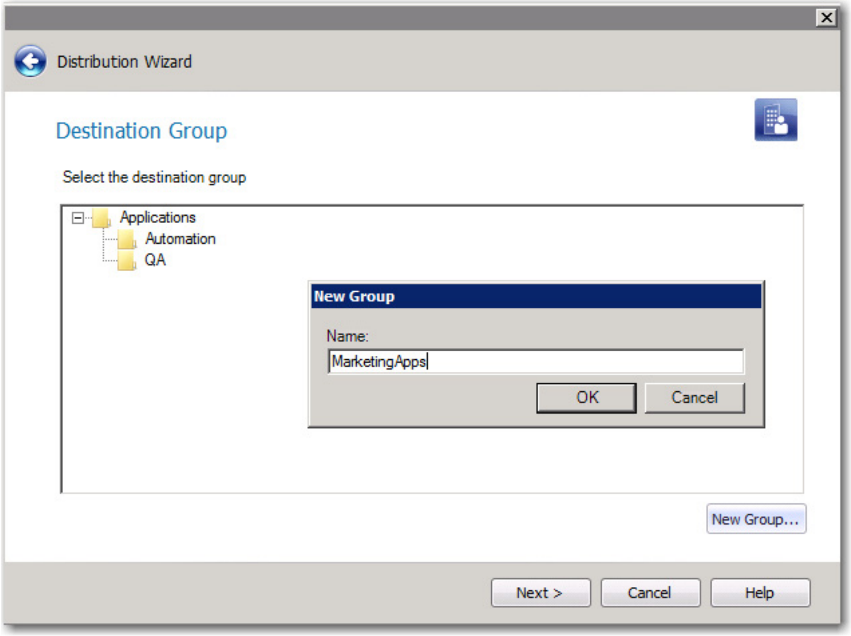
- System Center Configuration Manager (Current Branch) supports Apple mobile apps only for operating systems iOS7 or later.

Ability to Add Destination Group During System Center Configuration Manager Distribution

In previous releases, when distributing applications to System Center Configuration Manager using the Distribution Wizard, you were prompted to select an existing distribution group on the System Center Configuration Manager server to publish the applications to.

In AdminStudio 2016 SP2, you can now create a new destination group in System Center Configuration Manager directly from the **Destination Group** panel of the Distribution Wizard.

To create a new group, click the **New Group** button on the **Destination Group** panel and then enter a name for the new group in the **New Group** dialog box. AdminStudio will create that group on the specified System Center Configuration Manager server, and publish the applications to that group.



Distribution Wizard

Destination Group

Select the destination group

- Applications
  - Automation
  - QA

New Group

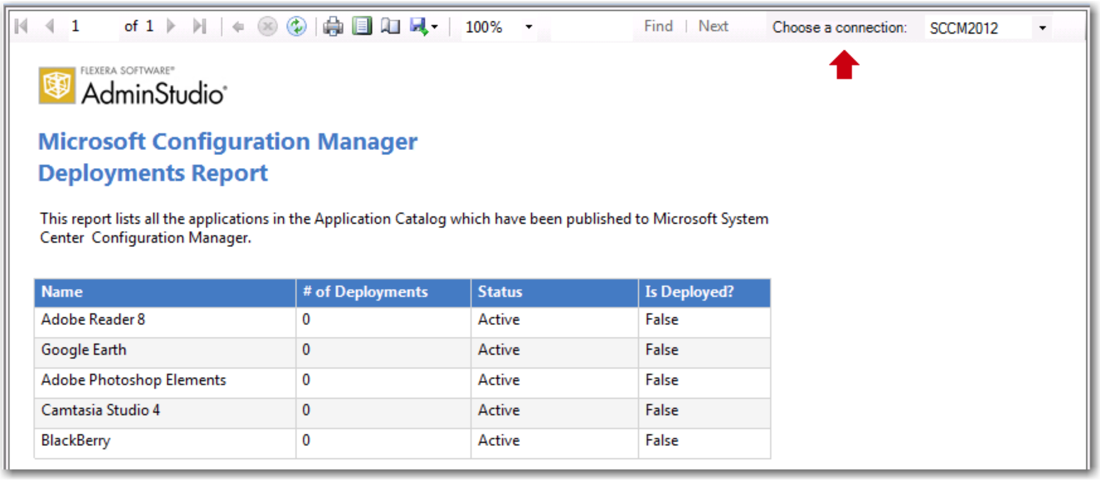
Name: MarketingApps

Figure 26: Adding a New Group on Destination Group Panel of Distribution Wizard

Ability to Select Connection on Microsoft Configuration Manager Deployments Report

When viewing the **Microsoft Configuration Manager Deployments Report** in AdminStudio 2016 SP2, you can now choose which System Center Configuration Manager connection that you want to view a report on.

If you have multiple connections to System Center Configuration Manager servers, select the connection that you want to view a report on from the **Choose a connection** list.



Microsoft Configuration Manager Deployments Report

This report lists all the applications in the Application Catalog which have been published to Microsoft System Center Configuration Manager.

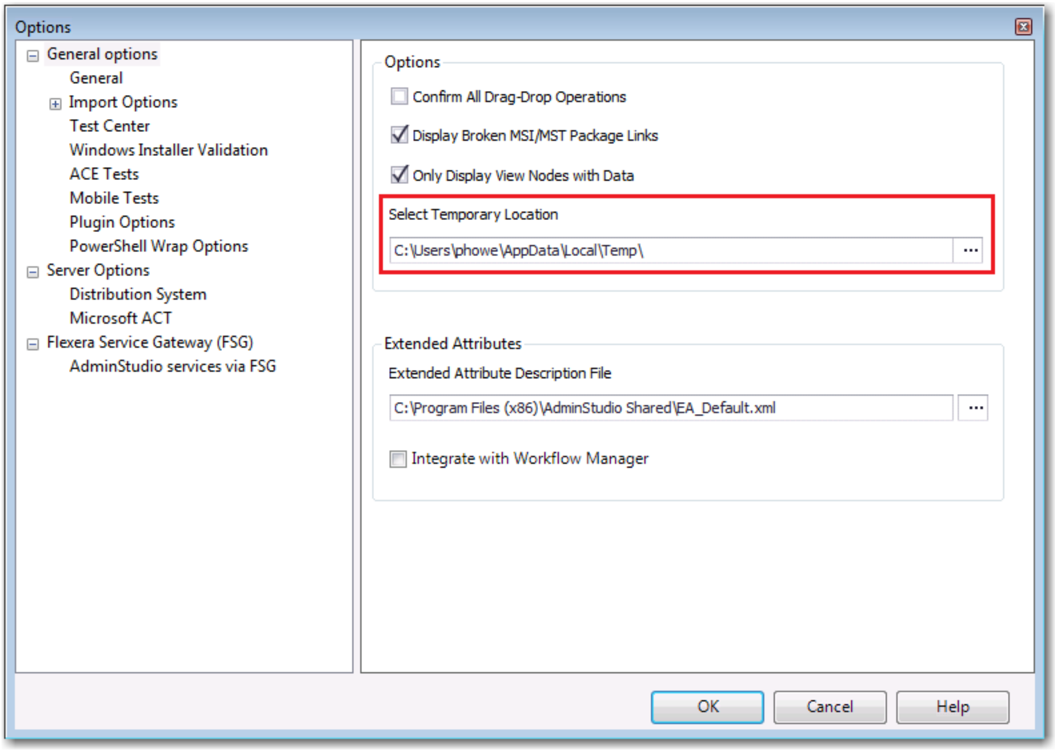
Name	# of Deployments	Status	Is Deployed?
Adobe Reader 8	0	Active	False
Google Earth	0	Active	False
Adobe Photoshop Elements	0	Active	False
Camtasia Studio 4	0	Active	False
BlackBerry	0	Active	False

Figure 27: Choosing a Connection on the Microsoft Configuration Manager Deployments Report

Ability to Set Temporary Directory Location

During import, packages are copied to an AdminStudio temporary directory to extract all of the data needed during import and testing. After package extraction, the files are deleted. In some cases, certain locations on a machine might be locked down and AdminStudio would be unable to access them, which would cause package extraction to fail.

Previously, you were unable to change the location of the AdminStudio temporary directory. In AdminStudio 2016 SP2, a new option, **Select Temporary Location**, has been added to the **General Options > General** tab of the Application Manager **Options** dialog box where you can specify the AdminStudio temporary directory. In this field, specify a location that AdminStudio has access to.



Options

□ Confirm All Drag-Drop Operations

☑ Display Broken MSI/MST Package Links

☑ Only Display View Nodes with Data

Select Temporary Location: C:\Users\phowe\AppData\Local\Temp\

Extended Attributes

Extended Attribute Description File: C:\Program Files (x86)\AdminStudio Shared\EA_Default.xml

□ Integrate with Workflow Manager

Figure 28: Select Temporary Location Option

Note • *This feature was tracked as issue IOJ-1799678.*

InstallShield 2016 SP2

AdminStudio 2016 SP2 includes InstallShield 2016 SP2. For a complete list of new features in InstallShield 2016 SP2, see the InstallShield 2016 SP2 Release Notes.

New Features in AdminStudio 2016 SP1

AdminStudio 2016 SP1 included the following new features:

- InstallShield 2016
- Support for the Microsoft UWP App Package (.appx) Format
- New API Commands to Set and Get Package Distribution Properties
- New OS Compatibility and Risk Assessment Test Center Tests
- Report Center Enhancements
- Applications Are Not Automatically Tested Upon Import
- Default ACE and ICE Tests are No Longer Specified on the ACE Tests Tab of Options Dialog Box

InstallShield 2016

AdminStudio 2016 SP1 includes InstallShield 2016. For a complete list of new features in InstallShield 2016, see the InstallShield 2016 SP2 Release Notes.

Support for the Microsoft UWP App Package (.appx) Format

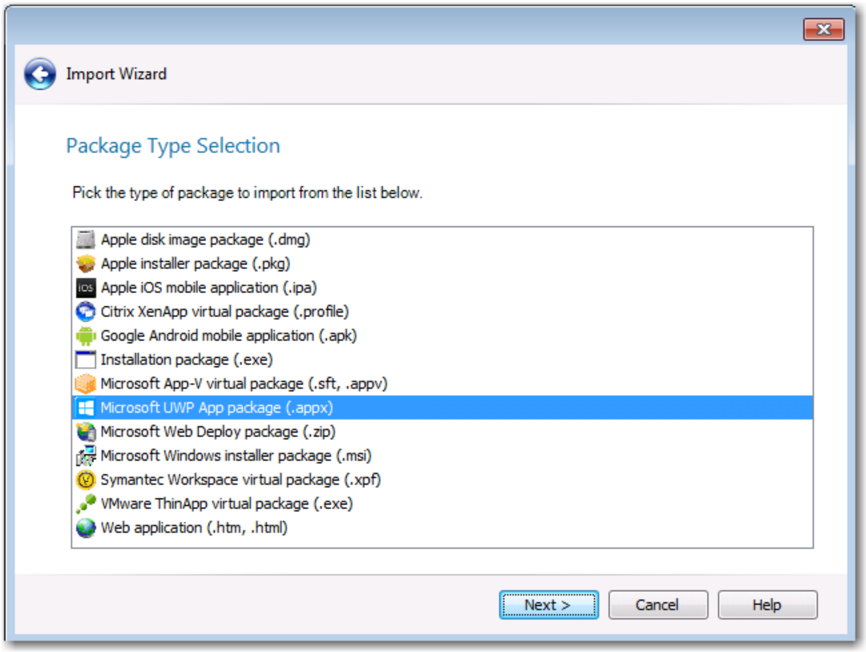
AdminStudio 2016 SP1 supports the import, testing, and distribution of Microsoft UWP app packages (`.appx`). The UWP app package format is the simple and secure packaging format used to distribute and install apps on Windows 8.x and 10 and is the only format allowed for Universal Windows Platform (UWP) apps. You have the option of selling or providing Microsoft UWP app packages through the Windows Store.

InstallShield 2016, also included with AdminStudio 2016 SP1, supports creating the UWP app package format (.appx) and its desktop and server extensions through an alternate build output and provides suitability testing to help you identify items unsuitable for the UWP app package format.

- Support for Importing UWP App Packages
- Support for Testing UWP App Packages
- Support for Distributing UWP App Packages to System Center Configuration Manager

Support for Importing UWP App Packages

You can import Microsoft UWP app packages using the Import Wizard by selecting the **Microsoft UWP App package (.appx)** option on the **Package Type Selection** panel of the **Import Wizard**. You can also import Microsoft UWP app packages from System Center Configuration Manager.



Import Wizard

Package Type Selection

Pick the type of package to import from the list below.

- Apple disk image package (.dmg)
- Apple installer package (.pkg)
- Apple iOS mobile application (.ipa)
- Citrix XenApp virtual package (.profile)
- Google Android mobile application (.apk)
- Installation package (.exe)
- Microsoft App-V virtual package (.sft, .appv)
- Microsoft UWP App package (.appx)
- Microsoft Web Deploy package (.zip)
- Microsoft Windows installer package (.msi)
- Symantec Workspace virtual package (.xpf)
- VMware ThinApp virtual package (.exe)
- Web application (.htm, .html)

Figure 29: Microsoft UWP App Package (.appx) Option on Package Type Selection Panel of Import Wizard

Support for Testing UWP App Packages

In AdminStudio 2016 SP1, you can use Application Manager Test Center to test imported Microsoft UWP app packages for operating system compatibility with Windows 8.1 and Window 10, and for mobile risk assessment.

Support for Distributing UWP App Packages to System Center Configuration Manager

In AdminStudio 2016 SP1, you can use the Distribution Wizard to distribute Microsoft UWP app packages to System Center 2012 Configuration Manager and later.

New API Commands to Set and Get Package Distribution Properties

AdminStudio 2016 SP1 includes two new API commands to set and get package distribution properties.

- **Set-ASDistributionProperty**—You can use the `Set-ASDistributionProperty` command to set the property value of the distribution system for a package. The following is an example of the `Set-ASDistributionProperty` command:

```
Set-ASDistributionProperty -PackageId 10 -SystemConnectionName 'Altiris'
    -PropertyName 'PackageName' -PropertyValue 'PackageABC'
```

- **Get-ASDistributionProperty**—You can use the `Get-ASDistributionProperty` command to retrieve the property value of the distribution system for a package. The following is an example of the `Get-ASDistributionProperty` command:

```
Get-ASDistributionProperty -PackageId 9 -SystemConnectionName 'Altiris' -PropertyName 'PackageName'
```

New OS Compatibility and Risk Assessment Test Center Tests

AdminStudio 2016 SP1 includes new tests for operating system compatibility with the latest versions of desktop and mobile operating systems. It also includes additional mobile risk assessment tests.

- New OS Compatibility Tests
- New Mobile Risk Assessment Tests

New OS Compatibility Tests

AdminStudio 2016 SP1 includes operating system compatibility tests for the following additional operating systems:

- Mac OS 10.12 (Sierra)
- Android 6.0 (Marshmallow) and Android 7.0 (Nougat)
- Apple iOS 9 and iOS 10
- Windows Server 2016

New Mobile Risk Assessment Tests

AdminStudio 2016 SP1 includes new mobile risk assessment tests for both iOS and Android operating systems.

New Apple Mobile Risk Assessment Tests

The following new mobile risk assessment tests were added for iOS mobile apps:

- 3D Touch (M153)
- ARM64 (M154)
- Siri Voice Assistance (SiriKit) (M155)

New Android Mobile Risk Assessment Tests

The following new mobile risk assessment tests were added for Android mobile apps:

- Android Pay (M260)
- Fingerprint Scanner (M261)
- Direct Share (M262)
- VR Mode (M263)

Report Center Enhancements

AdminStudio 2016 SP1 includes the following Report Center enhancements:

- New Devices in Device Compatibility Reports
- New Mac Desktop OS Application Compatibility Dashboard Report
- Mac Installer Best Practices Report

New Devices in Device Compatibility Reports

AdminStudio 2016 SP1's Device Compatibility reports in Report Center now include the following additional iOS devices:

New iOS Devices

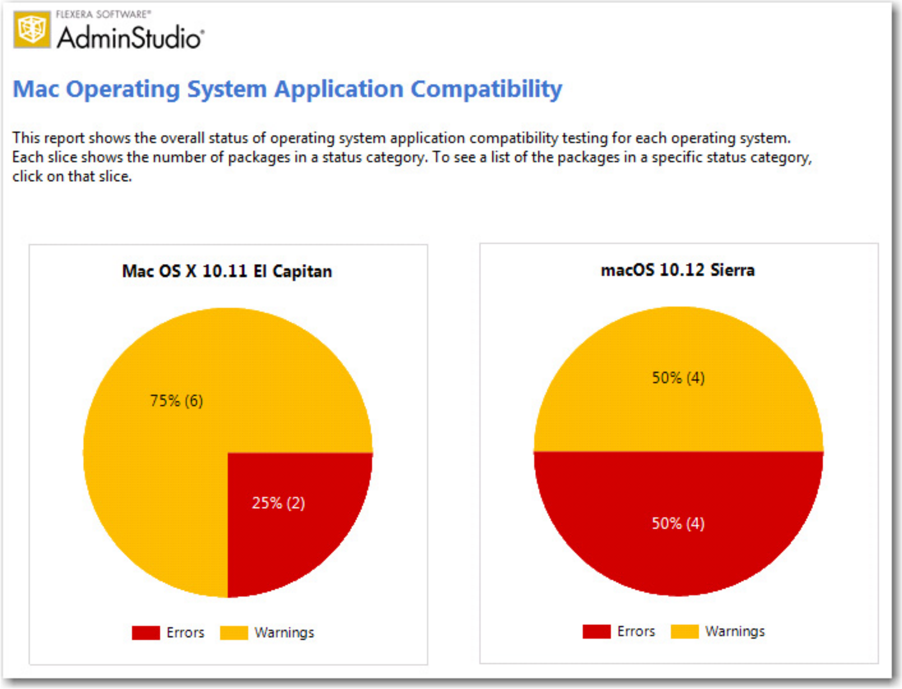
- iPad Pro 9.7 and 9.7 LTE
- iPad Pro 12.9 and 12.9 LTE
- iPad Mini 4 and 4 LTE
- iPhone 6s, 6s Plus, and SE

New Android Devices

- Google/LG Nexus 5X, 6P
- Samsung Galaxy Note 4, 5, 6, 7
- Samsung Galaxy S7
- Samsung Galaxy Tab S2

New Mac Desktop OS Application Compatibility Dashboard Report

AdminStudio 2016 SP1 includes a new Mac Operating System Application Compatibility dashboard report that shows the overall status of operating system application compatibility testing for Mac OS X 10.11 (El Capitan) and Mac OS 10.12 (Sierra).



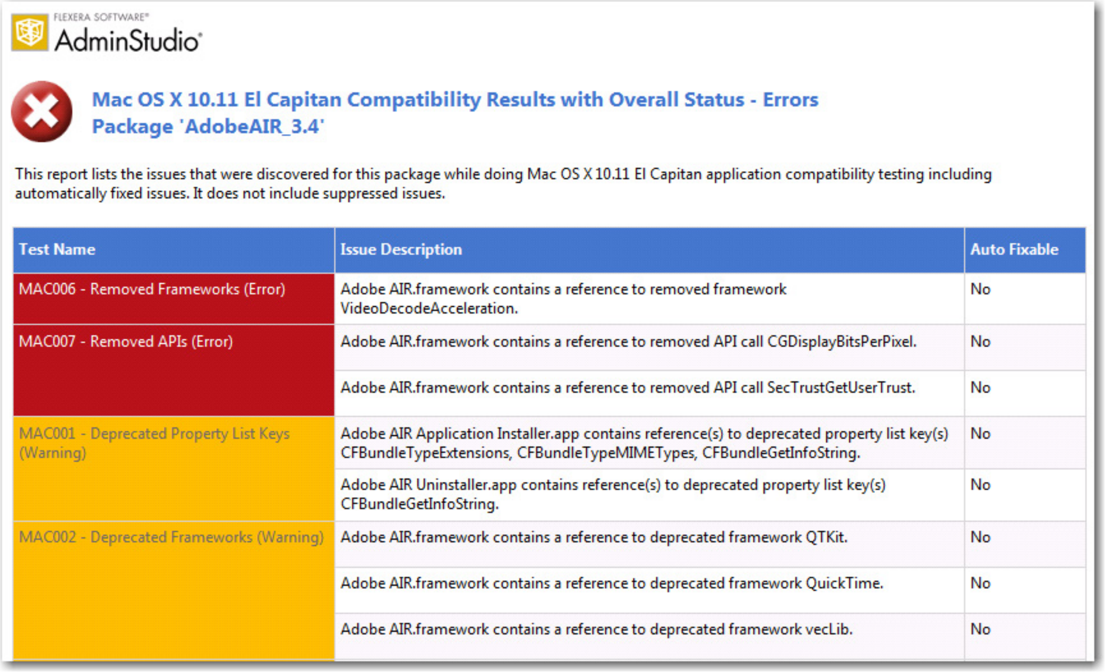
FLEXERA SOFTWARE® AdminStudio®

Mac Operating System Application Compatibility

This report shows the overall status of operating system application compatibility testing for each operating system. Each slice shows the number of packages in a status category. To see a list of the packages in a specific status category, click on that slice.

Figure 30: Mac OS Application Compatibility Dashboard

You can click on wedges of the pie charts in this dashboard report to drill down to view more detailed reports.



FLEXERA SOFTWARE®
AdminStudio®

Mac OS X 10.11 El Capitan Compatibility Results with Overall Status - Errors
Package 'AdobeAIR_3.4'

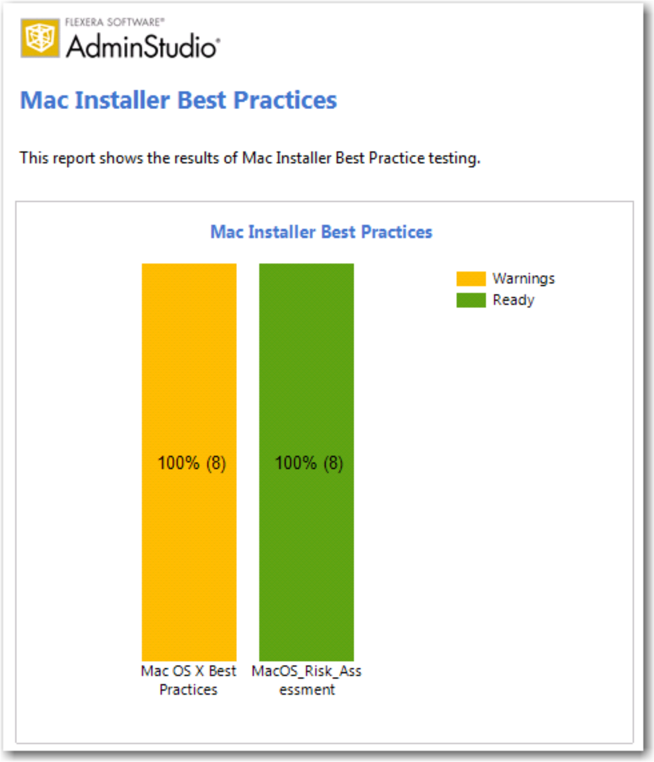
This report lists the issues that were discovered for this package while doing Mac OS X 10.11 El Capitan application compatibility testing including automatically fixed issues. It does not include suppressed issues.

Test Name	Issue Description	Auto Fixable
MAC006 - Removed Frameworks (Error)	Adobe AIR.framework contains a reference to removed framework VideoDecodeAcceleration.	No
MAC007 - Removed APIs (Error)	Adobe AIR.framework contains a reference to removed API call CGDisplayBitsPerPixel.	No
	Adobe AIR.framework contains a reference to removed API call SecTrustGetUserTrust.	No
MAC001 - Deprecated Property List Keys (Warning)	Adobe AIR Application Installer.app contains reference(s) to deprecated property list key(s) CFBundleTypeExtensions, CFBundleTypeMIMETypes, CFBundleGetInfoString.	No
	Adobe AIR Uninstaller.app contains reference(s) to deprecated property list key(s) CFBundleGetInfoString.	No
MAC002 - Deprecated Frameworks (Warning)	Adobe AIR.framework contains a reference to deprecated framework QTKit.	No
	Adobe AIR.framework contains a reference to deprecated framework QuickTime.	No
	Adobe AIR.framework contains a reference to deprecated framework vecLib.	No

Figure 31: Mac OS X 10.11 El Capitan Compatibility Results With Overall Status Report

Mac Installer Best Practices Report

AdminStudio 2016 SP1 includes a new Mac Installer Best Practices report that shows the results of Mac OS Installer best practices testing.



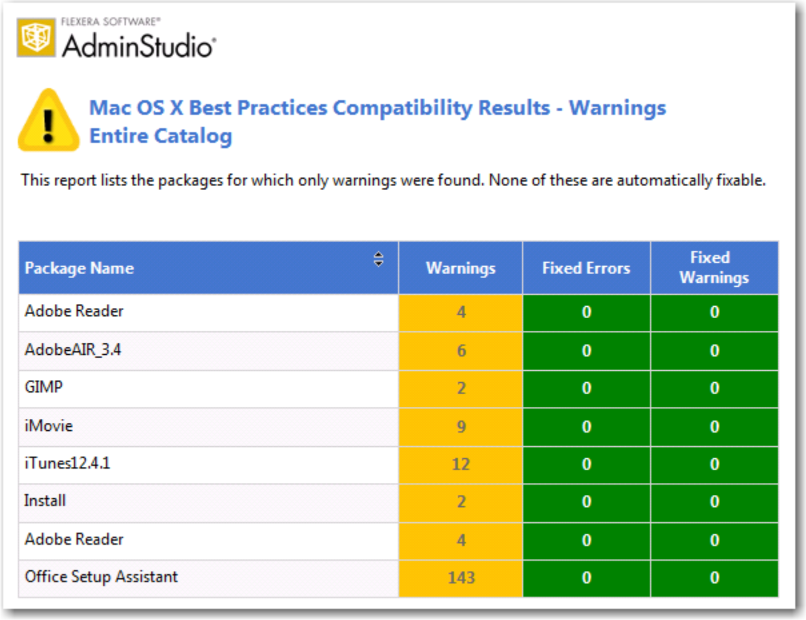
FLEXERA SOFTWARE® AdminStudio®

Mac Installer Best Practices

This report shows the results of Mac Installer Best Practice testing.

Figure 32: Mac Installer Best Practices Report

You can click on the bars in this report to drill down to view more detailed reports.



Mac OS X Best Practices Compatibility Results - Warnings
Entire Catalog

This report lists the packages for which only warnings were found. None of these are automatically fixable.

Package Name	Warnings	Fixed Errors	Fixed Warnings
Adobe Reader	4	0	0
AdobeAIR_3.4	6	0	0
GIMP	2	0	0
iMovie	9	0	0
iTunes12.4.1	12	0	0
Install	2	0	0
Adobe Reader	4	0	0
Office Setup Assistant	143	0	0

Figure 33: Mac OS X Best Practices Compatibility Results Report

Applications Are Not Automatically Tested Upon Import

In AdminStudio 2016, Application Manager was configured to automatically test packages against all selected compatibility, best practices, and risk assessment tests as part of the import process. By default, the **Automatically Execute Tests After Import** option on the **Import Options** tab of the Application Manager **Options** dialog box was selected.

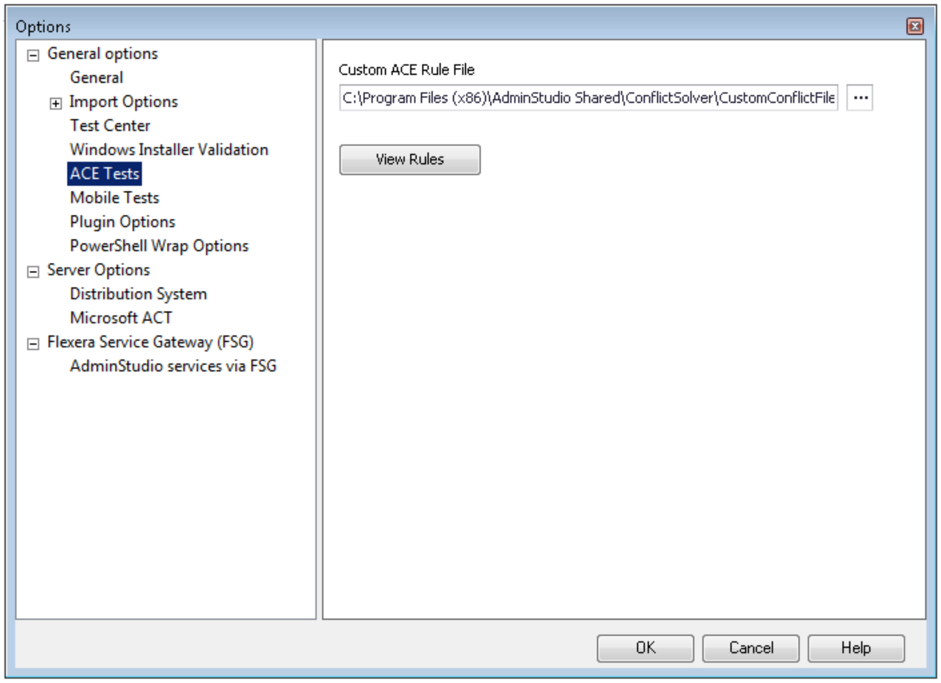
in AdminStudio 2016 SP1, the **Automatically Execute Tests After Import** option on the **Import Options** tab of the **Options** dialog box is no longer selected by default.

Default ACE and ICE Tests are No Longer Specified on the ACE Tests Tab of Options Dialog Box

In previous releases, you could specify the default ACE and ICE rules that you wanted to use in testing on the **ACE Tests** tab of the Application Manager **Options** dialog box. The selections that you made on the **ACE Tests** tab would be reflected on the **Select Tests to Execute** dialog box, accessible on the Test Center tab.

Starting with AdminStudio 2016 SP1, you no longer specify default ACE and ICE rules on the **ACE Tests** tab. Instead, you specify default ACE and ICE rules to run on the **Select Tests to Execute** dialog box only.

On the **ACE Tests** tab of the **Options** dialog box, you can still access the **Rules Viewer**—which can be used to add user defined ACEs—and you can still specify a custom ACE rule file.



Custom ACE Rule File

C:\Program Files (x86)\AdminStudio Shared\ConflictSolver\CustomConflictFile

Figure 34: ACE Tests Tab of Application Manager Options Dialog Box

New Features in AdminStudio 2016

AdminStudio 2016 included the following new features:

- Import and Test Mac OS X Desktop Applications
- Ability to Inspect and Test Complex Installer Executables
- Improved FlexNet Manager Suite Integration
- Enhanced Automation Capabilities to Streamline Application Import and Testing
- Automated Application Converter Enhancements
- New Wizard to Install and Launch Windows Installer Packages on a Virtual Machine for Testing
- Productivity Shortcuts
- New System Center Configuration Manager Settings View

Import and Test Mac OS X Desktop Applications

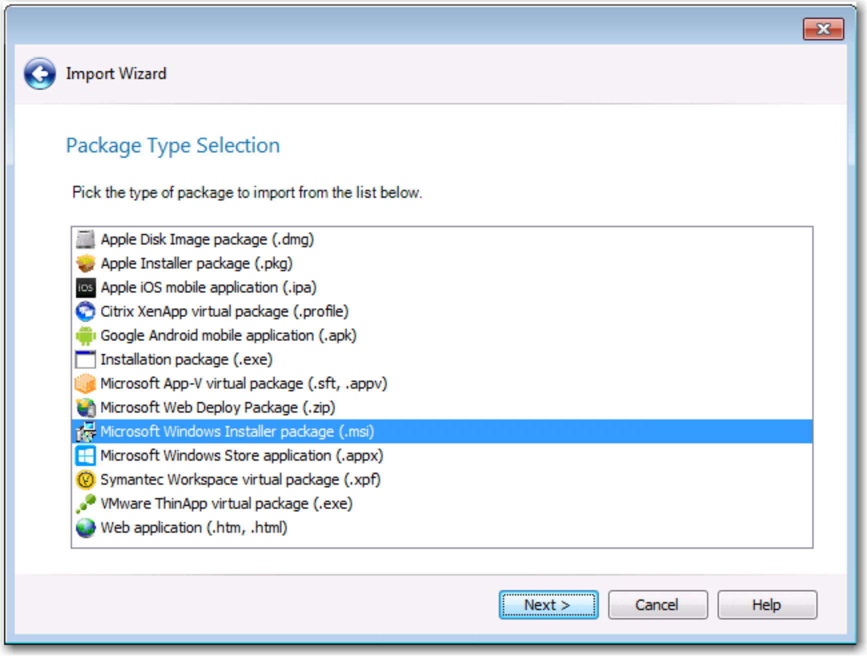
In its 2016 release, AdminStudio is extending its application readiness capabilities to Mac OS X desktop applications. You can now import Mac OS X applications into the Application Catalog for testing and distribution to a JAMF Casper Suite server. Additionally, because App Portal also supports Casper, when you publish DMG and PKG files to Casper, you can choose to automatically generate App Portal catalog items for those packages.

- Import Mac OS X Desktop Applications (DMG and PKG Files)
- Import Links to OS X Desktop Applications in the Mac App Store
- Test Mac OS X Desktop Applications
- Publish Mac OS X Desktop Applications to Casper With Integration to App Portal

Import Mac OS X Desktop Applications (DMG and PKG Files)

In AdminStudio 2016, you can import Apple disk images (`.dmg`) and Mac OS X installer packages (`.pkg`) into the Application Catalog for inspection, customization, testing, and distribution to Casper.

The **Package Type Selection** panel of the Import Wizard now includes two new selections to support the import of Mac OS X desktop applications into the Application Catalog.



Import Wizard

Package Type Selection

Pick the type of package to import from the list below.

- Apple Disk Image package (.dmg)
- Apple Installer package (.pkg)
- Apple iOS mobile application (.ipa)
- Citrix XenApp virtual package (.profile)
- Google Android mobile application (.apk)
- Installation package (.exe)
- Microsoft App-V virtual package (.sft, .appv)
- Microsoft Web Deploy Package (.zip)
- Microsoft Windows Installer package (.msi)
- Microsoft Windows Store application (.appx)
- Symantec Workspace virtual package (.xpf)
- VMware ThinApp virtual package (.exe)
- Web application (.htm, .html)

Figure 35: New Apple Installer Packages on Package Type Selection Panel

The following are the two new package types that were added to the **Package Type Selection** panel to support Mac OS X desktop applications:

- **Apple Disk Image package (.dmg)**—When you double-click a `.dmg` file, an Apple disk image is “mounted” as a volume within the Finder. To install the application, you usually drag the application icon from the disk image into the Applications folder.
- **Apple installer package (.pkg)**—Double-clicking a `.pkg` file launches the Apple installer application, where the package is installed by proceeding through an installation wizard.

You can import these newly supported package types as single applications, part of a batch import of applications in a selected folder, or as part of a monitored directory using the Package Auto Import feature. You can also publish these package types to a JAMF Casper Suite server.

Inspecting Bundled Packages

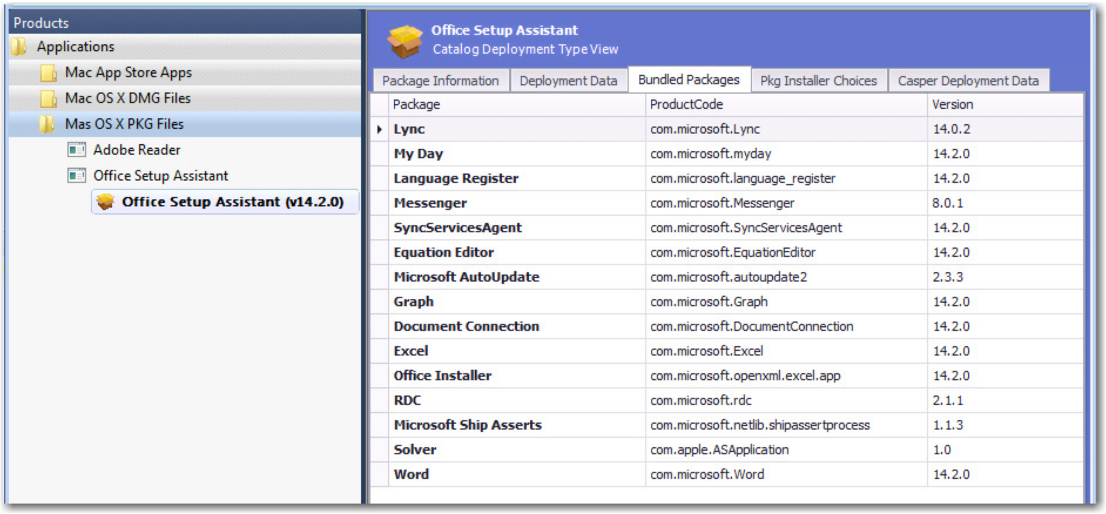
Both `.dmg` and `.pkg` files can contain child `.dmg` or `.pkg` files bundled within the file. These bundled packages can be viewed on the imported package’s **Bundled Packages** tab of the **Catalog Deployment Type View**. For more information, see Inspect Bundled Applications in DMG and PKG Installers.

Customizing Installer Settings

The installer settings of a **.pkg** file can be modified on the **Pkg Installer Choices** tab of the **Catalog Deployment Type View**. For more information, see Customize Apple Installer Package PKG Installer Settings.

Inspect Bundled Applications in DMG and PKG Installers

If an Apple disk image package (`.dmg`) or Apple installer package (`.pkg`) contains child packages bundled within it, those child packages will be listed on the new **Bundled Packages** tab of the **Catalog Deployment Type** view.



Package	ProductCode	Version
Lync	com.microsoft.Lync	14.0.2
My Day	com.microsoft.myday	14.2.0
Language Register	com.microsoft.language_register	14.2.0
Messenger	com.microsoft.Messenger	8.0.1
SyncServicesAgent	com.microsoft.SyncServicesAgent	14.2.0
Equation Editor	com.microsoft.EquationEditor	14.2.0
Microsoft AutoUpdate	com.microsoft.autoupdate2	2.3.3
Graph	com.microsoft.Graph	14.2.0
Document Connection	com.microsoft.DocumentConnection	14.2.0
Excel	com.microsoft.Excel	14.2.0
Office Installer	com.microsoft.openxml.excel.app	14.2.0
RDC	com.microsoft.rdc	2.1.1
Microsoft Ship Asserts	com.microsoft.netlib.shipassertprocess	1.1.3
Solver	com.apple.ASApplication	1.0
Word	com.microsoft.Word	14.2.0

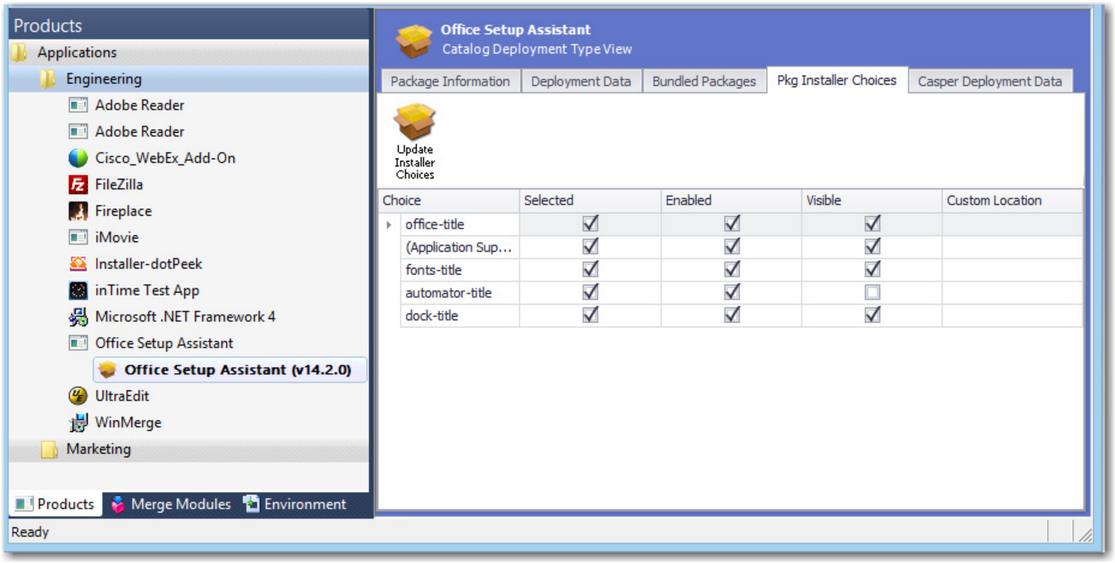
Figure 36: Bundled Packages Tab of Catalog Deployment Type View for Apple Installer Package (.pkg)

When the imported `.dmg` or `.pkg` file is tested, each of these child packages is also tested, and the combined test results are listed in Test Center, as described in Combined Testing of Child Applications of DMG and PKG Files.

***Note** • AdminStudio will only inspect Mac OS X package files one level deep. If a .dmg or .pkg package contains another .dmg or .pkg package bundled within it, that child package will not be inspected.*

Customize Apple Installer Package PKG Installer Settings

Just as a Windows Installer package can be customized by adding a transform file, an Apple installer package (`.pkg`) can be customized by editing an XML file that is embedded within it. In AdminStudio 2016, the settings defined in a `.pkg` file's embedded XML file are displayed on the new **PKG Installer Choices** tab of the package's **Catalog Deployment Type** view.



Choice	Selected	Enabled	Visible	Custom Location
office-title	☑	☑	☑	
(Application Sup…	☑	☑	☑	
fonts-title	☑	☑	☑	
automator-title	☑	☑	□	
dock-title	☑	☑	☑	

Figure 37: New PKG Installer Choices Tab of Catalog Deployment Type View for Mac PKG Installer

The **PKG Installer Choices** tab lists all settings that have been defined in that `.pkg` file's embedded XML settings file by the application manufacturer. To customize this installer (such as to prepare it for silent installation by Casper), you can make changes to the settings on this tab and then click **Update Installer Choices**. AdminStudio will then save your changes in the package's embedded settings file.

For each installer **Choice** listed on the **PKG Installer Choices** tab, the following options are available:

Option	Description
Visible	This option can be either selected or not selected: ● **Selected**—This choice setting will be displayed in the installer. ● **Not selected**—This choice setting will not be displayed in the installer.
Selected	This option can be either selected or not selected: ● **Selected**—If this choice setting is displayed in the installer, its check box will be checked. ● **Not selected**—If this choice setting is displayed in the installer, its check box will not be checked.
Enabled	This option can be either selected or not selected: ● **Selected**—If this choice setting is displayed in the installer, it will be enabled. ● **Not selected**—If this choice setting is displayed in the installer, it will be disabled.
Custom Location	If this choice setting explicitly permits the user to specify a user-defined installation path, the path entered in this field would populate the user-defined installation path when it is displayed in the installer.

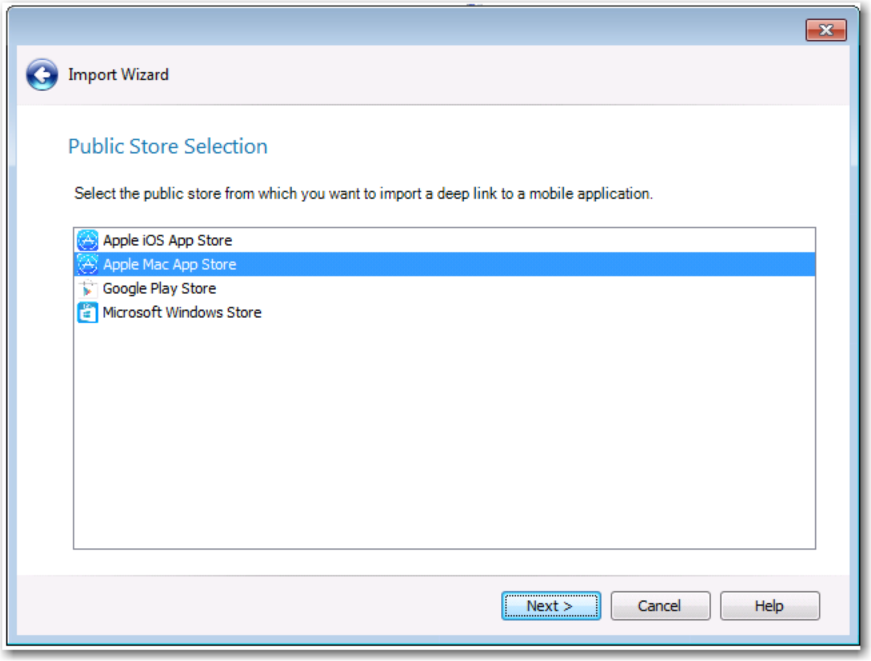
***Note •** Modifying the installer choices of an Apple installer package does not affect the digital signature of the package.*

Import Links to OS X Desktop Applications in the Mac App Store

In previous releases, you could import links to iOS mobile apps in the iTunes App Store. To do this, you selected, **Apple Store** on the **Public Store Selection** panel of the Import Wizard.

In AdminStudio 2016, in addition to being able to import links to iOS mobile apps in the iTunes App Store, you can now also import links to Mac OS X desktop applications in the Mac App Store. On the **Public Store Selection** panel of the Import Wizard, you now have two different Apple public store options:

- **Apple iOS App Store** (for iOS mobile apps)
- **Apple Mac App Store** (for OS X desktop apps)



Public Store Selection

Select the public store from which you want to import a deep link to a mobile application.

Apple iOS App Store
Apple Mac App Store
Google Play Store
Microsoft Windows Store

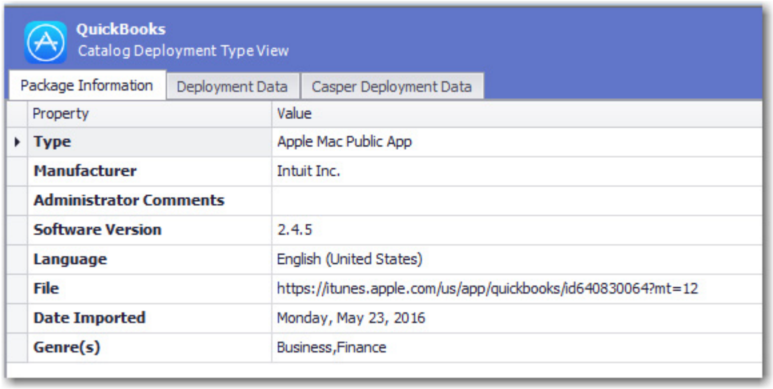
Figure 38: Apple Mac App Store Option on Public Store Selection Panel

After you have imported a Mac OS X desktop application into the Application Catalog, you can test it and distribute it to a Casper server. For more information, see:

- Test Mac OS X Desktop Applications
- Publish Mac OS X Desktop Applications to Casper With Integration to App Portal

New Field That Identifies Genre of Mac Public Store Apps (Both iOS and OS X)

For Apple public store apps imported into the Application Catalog (both iOS and OS X apps), a new **Genre(s)** field is listed on the **Package Information** tab of the **Catalog Deployment Type** View, which lists the category that the application belongs to in the public store.

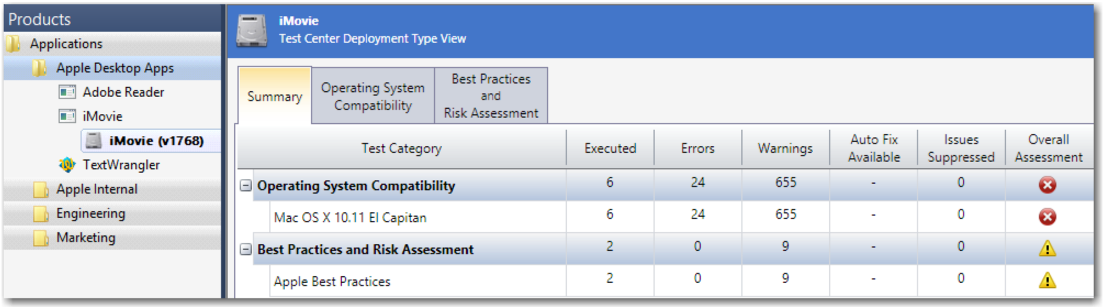


Property	Value
Type	Apple Mac Public App
Manufacturer	Intuit Inc.
Administrator Comments	
Software Version	2.4.5
Language	English (United States)
File	https://itunes.apple.com/us/app/quickbooks/id640830064?mt=12
Date Imported	Monday, May 23, 2016
Genre(s)	Business,Finance

Figure 39: Genre(s) Field on Package Information Tab

Test Mac OS X Desktop Applications

You can use Test Center to test all three types of supported Mac OS X desktop applications: DMG files, PKG files, and Mac App Store apps. Tests are available in the **Operating System Compatibility** and **Best Practices and Risk Assessment** test categories.



Test Category	Executed	Errors	Warnings	Auto Fix Available	Issues Suppressed	Overall Assessment
⊟ **Operating System Compatibility**	6	24	655	-	0	✖
Mac OS X 10.11 El Capitan	6	24	655	-	0	✖
⊟ **Best Practices and Risk Assessment**	2	0	9	-	0	⚠
Apple Best Practices	2	0	9	-	0	⚠

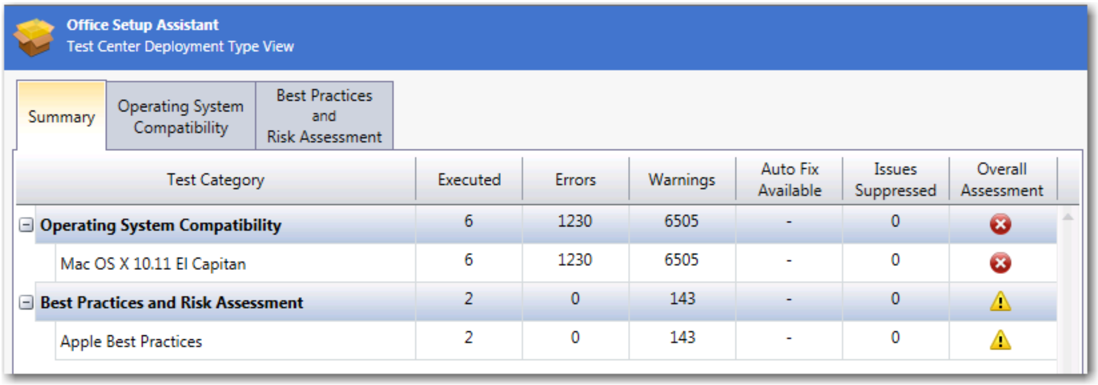
Figure 40: Test Results for Mac OS X Desktop App

In AdminStudio 2016, the following new tests for Mac OS X desktop applications have been added:

Category	Test Name	Description	Supported Package Types
Operating System Compatibility	MAC001	**Deprecated Property List Keys**	DMG, PKG
	MAC002	**Deprecated Frameworks**	DMG, PKG
	MAC003	**Application Requires Specific Minimum OS Version**	DMG, PKG, Mac App Store
	MAC004	**Deprecated APIs**	DMG, PKG
	MAC005	**Application Requires 64-bit Processor**	Mac App Store
	MAC006	**Removed Frameworks**	DMG, PKG
	MAC007	**Removed APIs**	DMG, PKG
Apple Best Practices	MAC701	**Recommended Property List Keys**	DMG, PKG
	MAC702	**Code Signature**	DMG, PKG
	MAC703	**Volume Purchase Program**	Mac App Store
	MAC704	**Allows In-App Purchases**	Mac App Store

Combined Testing of Child Applications of DMG and PKG Files

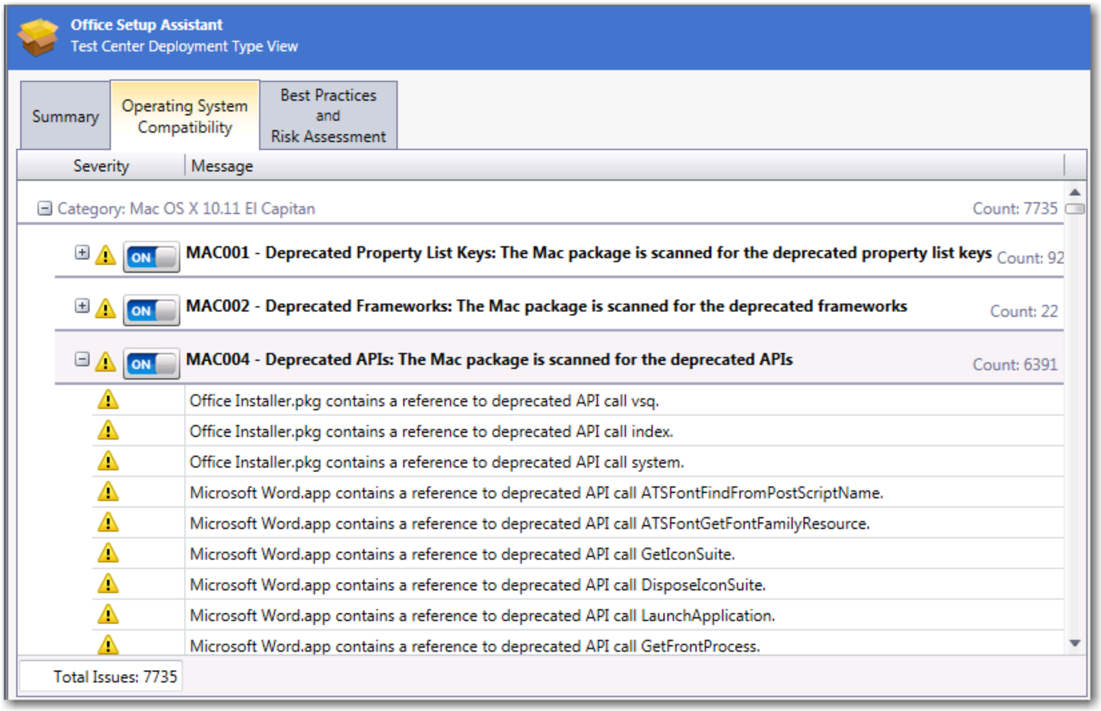
If an Apple disk image package (`.dmg`) or Apple installer package (`.pkg`) contains child packages bundled within it, those child packages are also tested when the parent package is tested. Test results of the parent package and all of its child packages are combined and are displayed in Test Center.


Office Setup Assistant
Test Center Deployment Type View
Summary
Operating System Compatibility
Best Practices and Risk Assessment

Test Category	Executed	Errors	Warnings	Auto Fix Available	Issues Suppressed	Overall Assessment
⊟ **Operating System Compatibility**	6	1230	6505	-	0	✖
Mac OS X 10.11 El Capitan	6	1230	6505	-	0	✖
⊟ **Best Practices and Risk Assessment**	2	0	143	-	0	⚠
Apple Best Practices	2	0	143	-	0	⚠

Figure 41: Consolidated Test Results for Apple Installer Package (.pkg)

When you view detailed test results, the name of the child `.dmg`, `.pkg`, or `.app` file that generated the error or warning is listed.



Office Setup Assistant
Test Center Deployment Type View
Summary
Operating System Compatibility
Best Practices and Risk Assessment
Total Issues: 7735

Severity	Message
⊟ Category: Mac OS X 10.11 El Capitan	Count: 7735
⊞ ⚠ ON	MAC001 - Deprecated Property List Keys: The Mac package is scanned for the deprecated property list keys Count: 92
⊞ ⚠ ON	MAC002 - Deprecated Frameworks: The Mac package is scanned for the deprecated frameworks Count: 22
⊟ ⚠ ON	MAC004 - Deprecated APIs: The Mac package is scanned for the deprecated APIs Count: 6391
⚠	Office Installer.pkg contains a reference to deprecated API call vsq.
⚠	Office Installer.pkg contains a reference to deprecated API call index.
⚠	Office Installer.pkg contains a reference to deprecated API call system.
⚠	Microsoft Word.app contains a reference to deprecated API call ATSFontFindFromPostScriptName.
⚠	Microsoft Word.app contains a reference to deprecated API call ATSFontGetFontFamilyResource.
⚠	Microsoft Word.app contains a reference to deprecated API call GetIconSuite.
⚠	Microsoft Word.app contains a reference to deprecated API call DisposeIconSuite.
⚠	Microsoft Word.app contains a reference to deprecated API call LaunchApplication.
⚠	Microsoft Word.app contains a reference to deprecated API call GetFrontProcess.

Figure 42: Detailed Consolidated Test Results for Apple Installer Package (.pkg)

Publish Mac OS X Desktop Applications to Casper With Integration to App Portal

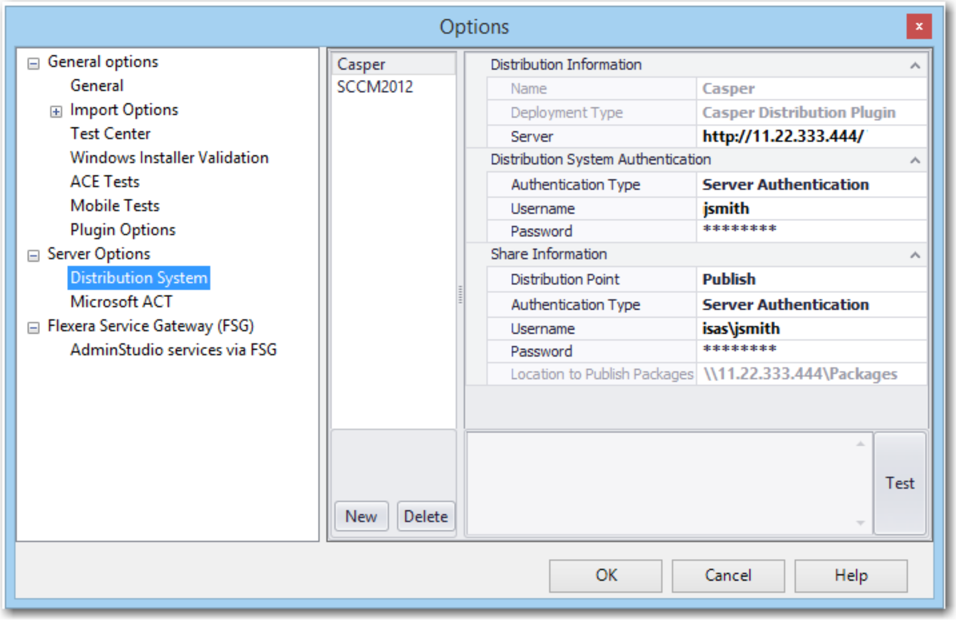
In AdminStudio 2016, you can publish Mac OS X desktop applications (DMG, PKG, and Mac App Store apps) to a JAMF Casper Suite server.

Because AdminStudio is also integrated with App Portal (which also supports Casper), you can choose to automatically create App Portal catalog items for DMG and PKG applications in your Application Catalog when you publish them to a Casper server.

- Setting Up a Connection to a Casper Server
- Automatically Creating App Portal Catalog Items for DMG and PKG Applications
- Viewing Casper Deployment Data for an Application

Setting Up a Connection to a Casper Server

On the **Server Options > Distribution Options** tab of the Application Manager **Options** dialog box, you can now create a named connection to a Casper server.



Distribution Information	
Name	Casper
Deployment Type	Casper Distribution Plugin
Server	http://11.22.333.444/
Distribution System Authentication	
Authentication Type	Server Authentication
Username	jsmith
Password	********
Share Information	
Distribution Point	Publish
Authentication Type	Server Authentication
Username	isas\jsmith
Password	********
Location to Publish Packages	\\11.22.333.444\Packages

Figure 43: Named Connection to a Casper Server

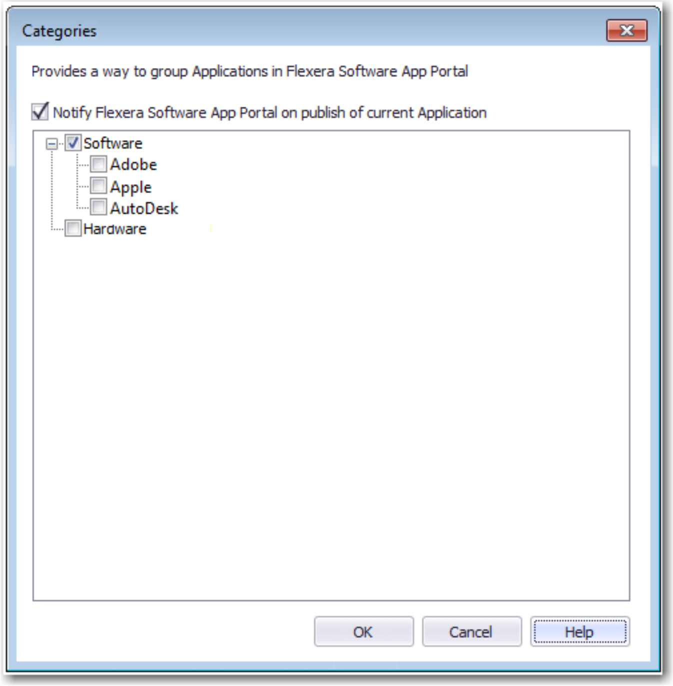
After you select **Casper Distribution Plugin** from the **Deployment Type** field, the settings required for connecting to a Casper server are listed.

***Note •** Casper supports multiple server infrastructures, but AdminStudio only supports the **File Share Distribution Points** infrastructure, and copies packages to a UNC File Share Distribution Point in Casper. AdminStudio currently does not support copying packages to JDS Instances, Cloud Distribution Points, Software Update Servers, or NetBoot Servers.*

Automatically Creating App Portal Catalog Items for DMG and PKG Applications

When you distribute DMG and PKG applications to Casper Suite Server, you can instruct AdminStudio to automatically create App Portal catalog items for each of those applications.

To do this, first set up a named connection to a Casper server and connect to the Flexera Service Gateway. Then, open the **App Portal Information** tab of the DMG or PKG application's **Application View**, and click on the **Categories** field to open the **Categories** dialog box. On the **Categories** dialog box, select the **Notify Flexera Software App Portal on publish of current application** option and select one or more App Portal categories where the catalog item will be available.


Categories

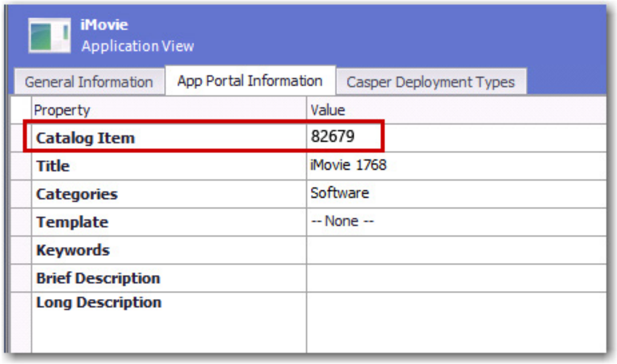
Provides a way to group Applications in Flexera Software App Portal

☑ Notify Flexera Software App Portal on publish of current Application

- ☑ Software
  - □ Adobe
  - □ Apple
  - □ AutoDesk
- □ Hardware

Figure 44: Selecting App Portal Options on Categories Dialog Box

When this application is published to Casper, a catalog item in the selected category will automatically be created, and the **Catalog Item** field on the **App Portal Information** view will display the App Portal Catalog ID for that catalog item.


iMovie
Application View

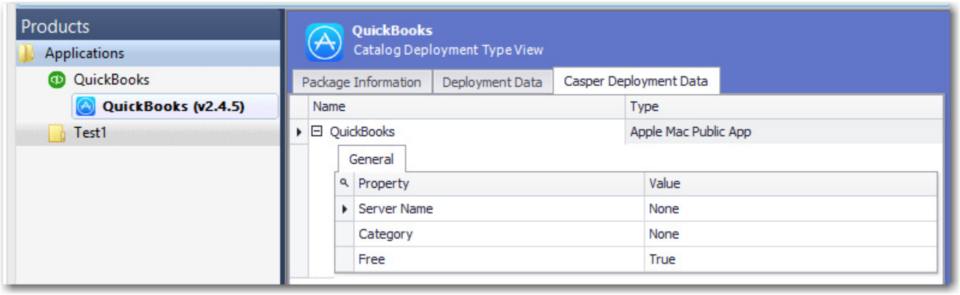
Property	Value
Catalog Item	82679
Title	iMovie 1768
Categories	Software
Template	-- None --
Keywords	
Brief Description	
Long Description	

Figure 45: Catalog Item Field on App Portal Information Tab

***Note •** If an App Portal catalog item has not been created for an application, the **Catalog Item** field will display* `Not Published to App Portal`.

Viewing Casper Deployment Data for an Application

In AdminStudio 2016, you can view and modify Casper deployment settings for Mac OS X desktop applications on the new **Casper Deployment Data** tab of the **Catalog Deployment Type View**.



Name	Type
QuickBooks	Apple Mac Public App

Property	Value
Server Name	None
Category	None
Free	True

Figure 46: Casper Deployment Data Tab of Catalog Deployment Type View

The **Casper Deployment Data** tab can include up to three subtabs that display Casper deployment data: **General**, **Options**, and **Limitations**.

***Note •** The **Options** and **Limitations** subtabs are not displayed for Mac App Store apps.*

The **Casper Deployment Data** tab displays the following properties:

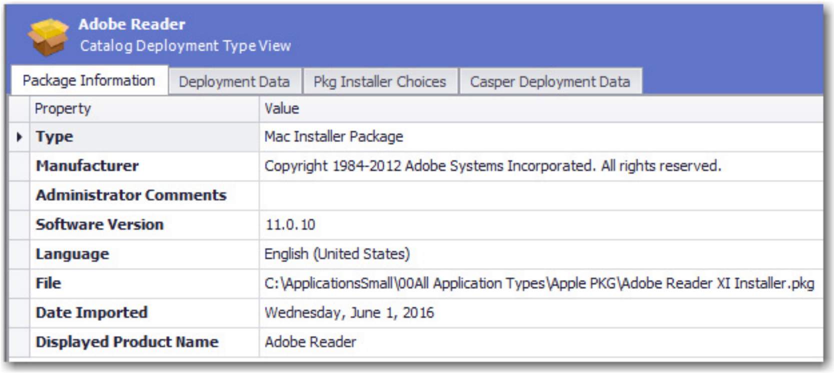
Subtab	Property	Description
General	**Server Name**	Name of the Casper server.
	Category	Category in Casper that the package will be added to. ***Note*** • *Casper lets you create custom categories. If AdminStudio has matched this application to an entry in the Application Recognition Library (ARL), AdminStudio will use the ARL category when publishing to Casper, creating it if necessary.*
	Info	Information to display to the administrator when the package is deployed or uninstalled
	Notes	Notes to display about the package (such as the name of the person who built it and when it was built). ***Note*** • *Not displayed for Mac App Store apps.*
	Free	Indicates whether or not the Mac App Store app is available for free (**True**) or whether it requires payment (**False**). ***Note*** • *Only displayed for Mac App Store apps.*

Subtab	Property	Description
Options	**Priority**	Priority to use for deploying or uninstalling the package. For example, a package with a priority of **1** is deployed or uninstalled before other packages. When several applications are deployed together, the one with the highest priority is installed first. Therefore, if one application requires that another application be installed first before it can be successfully installed, you should assign the required application a higher priority (lower number) than the dependent application.
	Fill user templates (FUT)	Set this property to **True** to fill new home directories with the contents of the home directory in the package's `Users` folder. This setting can be changed when deploying or uninstalling the package using a policy.l ***Note** • Only applicable to DMG packages.*
	Fill existing user home directories (FEU)	Set this property to **True** to fill existing home directories with the contents of the home directory in the package's **Users** folder. This setting can be changed when deploying or uninstalling the package using a policy ***Note** • Only applicable to DMG packages.*
	Requires restart	Set this property to **True** to require that computers must be restarted after installing the package.
	Install on boot drive after imaging	Set this property to **True** to ensure that the package is installed on the boot drive after imaging. ***Note** • This setting is only used when deploying a package with an OS image, like with an OSD. It does not affect day-to-day package delivery.*

Subtab	Property	Description
Limitations	**Operating System Requirement**	Enter operating system version numbers, separated by commas, to specify that the package only be permitted to be deployed to computers with these operating system versions. To restrict installation to OS X 10.6.8, 10.7.x, or 10.8, you would enter the following: `10.6.8, 10.7.x, 10.8`
	Install Only if Available in Software Update	Set to **True** to require that this package only be installed if it is available in a software update.
	Limit Architecture Type	Set to **True** to require that this package only be installed on machines matching the selected **Architecture Type Requirement**.
	Architecture Type Requirement	If **Limit Architecture Type** is set to **True**, select the one of the following to specify the architecture type required to deploy the package: ● PowerPC ● Intel/X86
	Substitute Package	If you want to specify a different package to deploy to computers that do not meet the architecture type requirement, click the Browse button in this field to open the **Select Substitute Package** dialog box, and select a substitute package from either Casper or the Application Catalog.

Product Name Displayed in Casper Listed on the Catalog Deployment Type View

For DMG and PKG packages imported into the Application Catalog, a **Displayed Product Name** field is listed on the **Package Information** tab of the **Catalog Deployment Type** View, listing the property that is mapped to the **File Name** property in Casper.



Property	Value
Type	Mac Installer Package
Manufacturer	Copyright 1984-2012 Adobe Systems Incorporated. All rights reserved.
Administrator Comments	
Software Version	11.0.10
Language	English (United States)
File	C:\ApplicationsSmall\00All Application Types\Apple PKG\Adobe Reader XI Installer.pkg
Date Imported	Wednesday, June 1, 2016
Displayed Product Name	Adobe Reader

Figure 47: Displayed Product Name Field on Package Information Tab

Ability to Inspect and Test Complex Installer Executables

In previous releases, you could import complex installer executable files (.exe) that contain bundled Windows Installer packages into the Application Catalog, but you could not view the names of those bundled packages and you could not test those packages. You were limited to just viewing general package information and deployment data.

In AdminStudio 2016, you can now view the names of bundled Windows Installer packages and perform operating system compatibility, application virtualization compatibility, and best practices testing on those bundled packages.

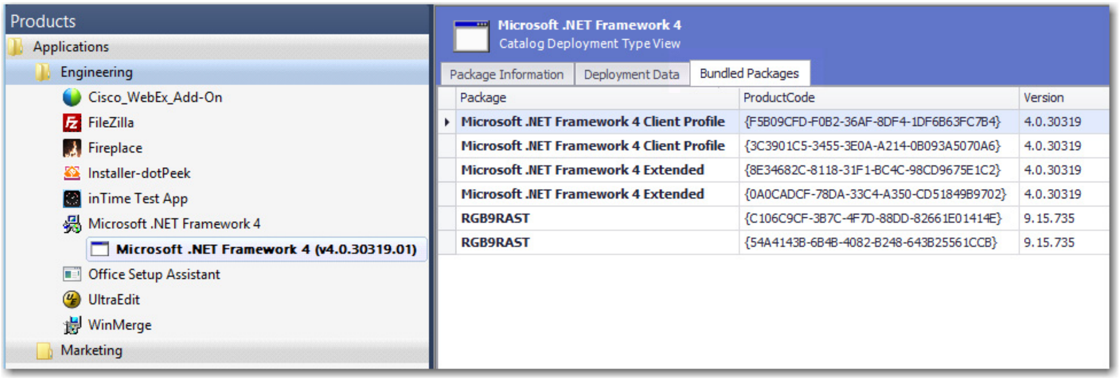
- Bundled Windows Installer Packages Now Listed on New Bundled Packages Tab
- Can Perform Testing of Child Windows Installer Packages of Complex Installer Executables

Bundled Windows Installer Packages Now Listed on New Bundled Packages Tab

There are multiple installation executable types that can contain embedded Windows Installer packages, including the following:

- InstallShield InstallScript `.exe` files
- InstallShield Basic MSI installers that are compressed into a `setup.exe` file
- InstallShield Suite Installer `.exe` files
- Wise Package Studio `.exe` files
- Other executable file types that can be uncompressed by 7-ZIP

In AdminStudio 2016, Application Manager can inspect the child `.msi` packages in these complex installer `.exe` files, and display a list of those child packages on the new **Bundled Packages** tab of the **Catalog Deployment Type View**.



Package	ProductCode	Version
Microsoft .NET Framework 4 Client Profile	{F5B09CFD-F0B2-36AF-8DF4-1DF6B63FC7B4}	4.0.30319
Microsoft .NET Framework 4 Client Profile	{3C3901C5-3455-3E0A-A214-0B093A5070A6}	4.0.30319
Microsoft .NET Framework 4 Extended	{8E34682C-8118-31F1-BC4C-98CD9675E1C2}	4.0.30319
Microsoft .NET Framework 4 Extended	{0A0CADCF-78DA-33C4-A350-CD51849B9702}	4.0.30319
RGB9RAST	{C106C9CF-3B7C-4F7D-88DD-82661E01414E}	9.15.735
RGB9RAST	{54A4143B-6B4B-4082-B248-643B25561CCB}	9.15.735

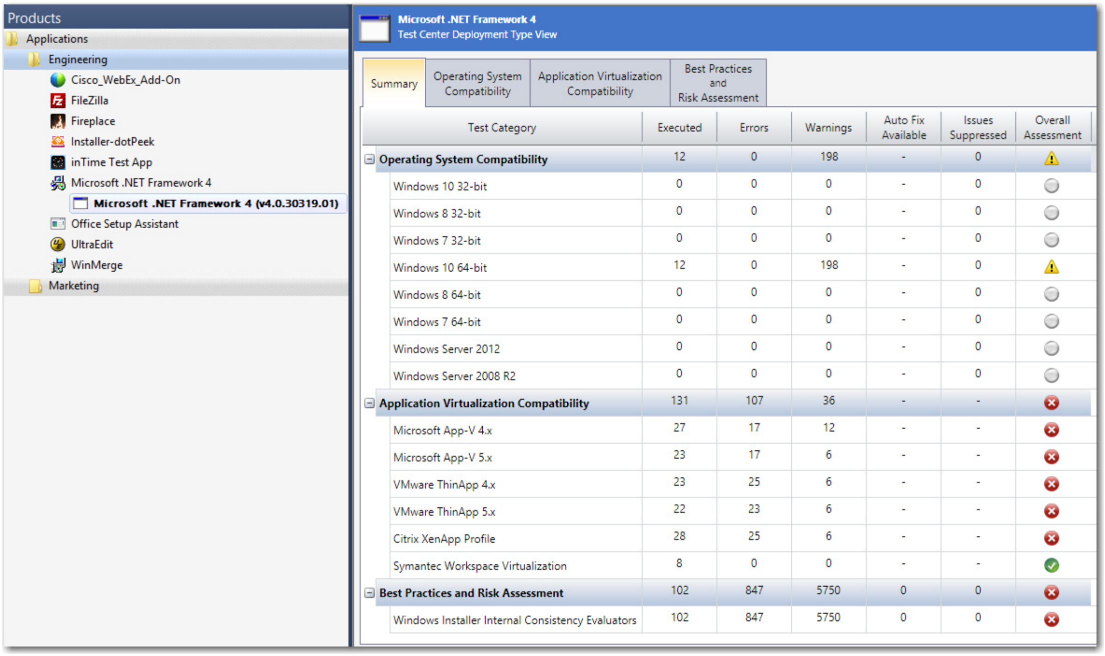
Figure 48: Bundled Packages Tab of Catalog Deployment Type View / Microsoft .NET Framework 4 Suite

When inspecting these child `.msi` packages, Application Manager extracts the information about each package, such as product name and version number. This makes it much more likely that Application Manager will be able to assign a Flexera Identifier to these applications.

Note • *AdminStudio will only inspect complex installer .exe files one level deep. If a complex installer .exe file contains another complex installer .exe file bundled within it, that child .exe file will not be inspected.*

Can Perform Testing of Child Windows Installer Packages of Complex Installer Executables

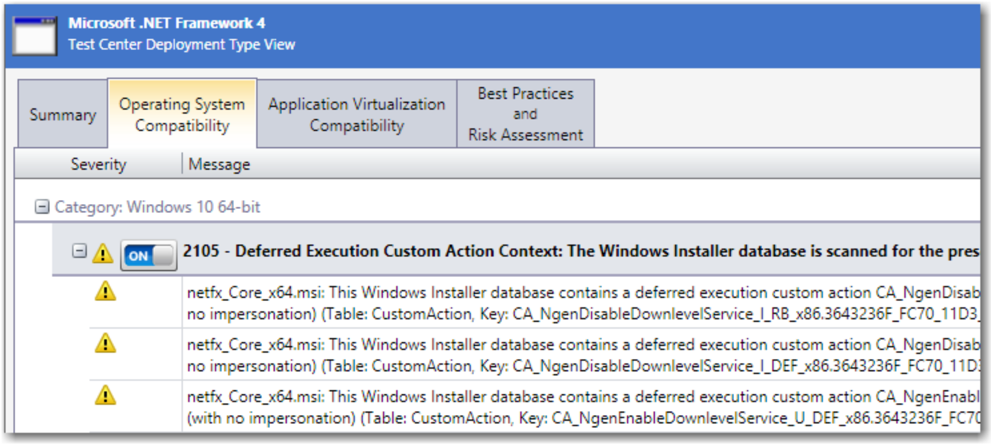
When a complex installer **.exe** file is tested, its child Windows Installer packages are also tested and the test results are combined and displayed in Test Center.


Products
Applications
Engineering
Cisco_WebEx_Add-On
FileZilla
Fireplace
Installer-dotPeek
inTime Test App
Microsoft .NET Framework 4
Microsoft .NET Framework 4 (v4.0.30319.01)
Office Setup Assistant
UltraEdit
WinMerge
Marketing
Microsoft .NET Framework 4
Test Center Deployment Type View
Summary
Operating System Compatibility
Application Virtualization Compatibility
Best Practices and Risk Assessment

Test Category	Executed	Errors	Warnings	Auto Fix Available	Issues Suppressed	Overall Assessment
⊟ **Operating System Compatibility**	12	0	198	-	0	⚠
Windows 10 32-bit	0	0	0	-	0	○
Windows 8 32-bit	0	0	0	-	0	○
Windows 7 32-bit	0	0	0	-	0	○
Windows 10 64-bit	12	0	198	-	0	⚠
Windows 8 64-bit	0	0	0	-	0	○
Windows 7 64-bit	0	0	0	-	0	○
Windows Server 2012	0	0	0	-	0	○
Windows Server 2008 R2	0	0	0	-	0	○
⊟ **Application Virtualization Compatibility**	131	107	36	-	-	✖
Microsoft App-V 4.x	27	17	12	-	-	✖
Microsoft App-V 5.x	23	17	6	-	-	✖
VMware ThinApp 4.x	23	25	6	-	-	✖
VMware ThinApp 5.x	22	23	6	-	-	✖
Citrix XenApp Profile	28	25	6	-	-	✖
Symantec Workspace Virtualization	8	0	0	-	-	✓
⊟ **Best Practices and Risk Assessment**	102	847	5750	0	0	✖
Windows Installer Internal Consistency Evaluators	102	847	5750	0	0	✖

Figure 49: Consolidated Test Results for Windows Installer Suite Package

When you view detailed test results, the name of the child Windows Installer package that generated the error or warning is listed.


Microsoft .NET Framework 4
Test Center Deployment Type View
Summary
Operating System Compatibility
Application Virtualization Compatibility
Best Practices and Risk Assessment
Severity
Message
Category: Windows 10 64-bit
ON

2105 - Deferred Execution Custom Action Context: The Windows Installer database is scanned for the pres

netfx_Core_x64.msi: This Windows Installer database contains a deferred execution custom action CA_NgenDisab
no impersonation) (Table: CustomAction, Key: CA_NgenDisableDownlevelService_I_RB_x86.3643236F_FC70_11D3_

netfx_Core_x64.msi: This Windows Installer database contains a deferred execution custom action CA_NgenDisab
no impersonation) (Table: CustomAction, Key: CA_NgenDisableDownlevelService_I_DEF_x86.3643236F_FC70_11D

netfx_Core_x64.msi: This Windows Installer database contains a deferred execution custom action CA_NgenEnabl
(with no impersonation) (Table: CustomAction, Key: CA_NgenEnableDownlevelService_U_DEF_x86.3643236F_FC70

Figure 50: Combined Suite Installer Test Results

Improved FlexNet Manager Suite Integration

In AdminStudio 2016, several new features have been added to improve integration with FlexNet Manager Suite. You can now generate a list of all applications in the Application Catalog that do not have an associated Flexera Identifier, and can create local Application Recognition Library entries for internally developed, repackaged, or unrecognized applications.

- Ability to Search Application Catalog for Unrecognized Applications
- Ability to Create Local Flexera Identifier Entries for Internal or Repackaged Applications
- Ability to Enter FlexNet Manager Suite URL

Ability to Search Application Catalog for Unrecognized Applications

A Flexera Identifier is a unique code, stored in the Application Recognition Library, that is assigned to applications by FlexNet Manager Suite. It is used to link application information from Application Manager with application information in AdminStudio Inventory and Rationalization, FlexNet Manager Suite, and App Portal.

If both Application Manager and FlexNet Manager Suite are connected to the same Flexera Service Gateway, each time you import an application into the Application Catalog, a search for the application’s Flexera Identifier is performed, and if it is found, it is listed on the **General Information** tab of the **Application View.**

However, sometimes an application’s Flexera Identifier is not found, such as when:

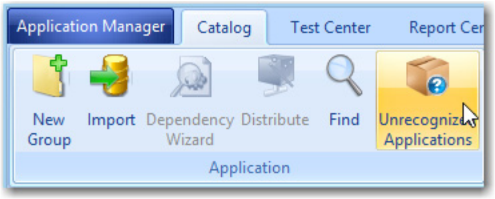
- **Incorrect information**—The value of the information in the application’s Product Name, Version, Edition, or Publisher fields is either incorrect or too specific.
- **Internally developed applications**—The application has been developed internally.
- **Repackaged applications**—The application has been repackaged.

In previous releases, there was no way to quickly identify all of the applications in your Application Catalog that did not have an assigned Flexera Identifier. You were required to review each application one by one.

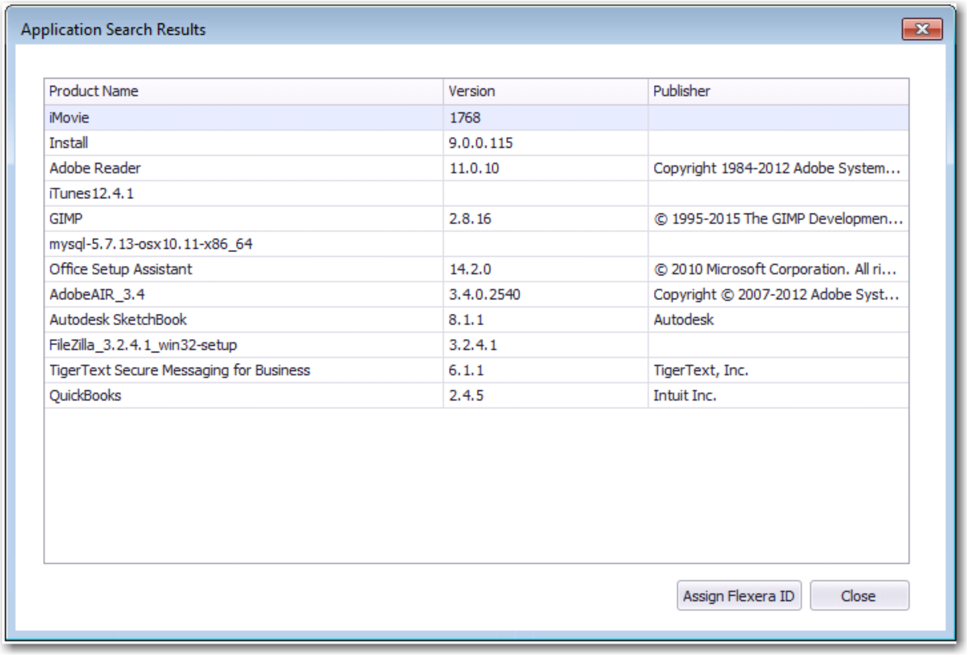
In AdminStudio 2016, you can now generate a list of all applications in the Application Catalog that do not have an associated Flexera Identifier, so that you can search for and assign an existing Flexera Identifier to the application or create a new local Flexera Identifier.

Task ***To search Application Catalog for unrecognized applications:***

1. Click **Unrecognized Applications** in the toolbar of the Application Manager **Catalog** tab.

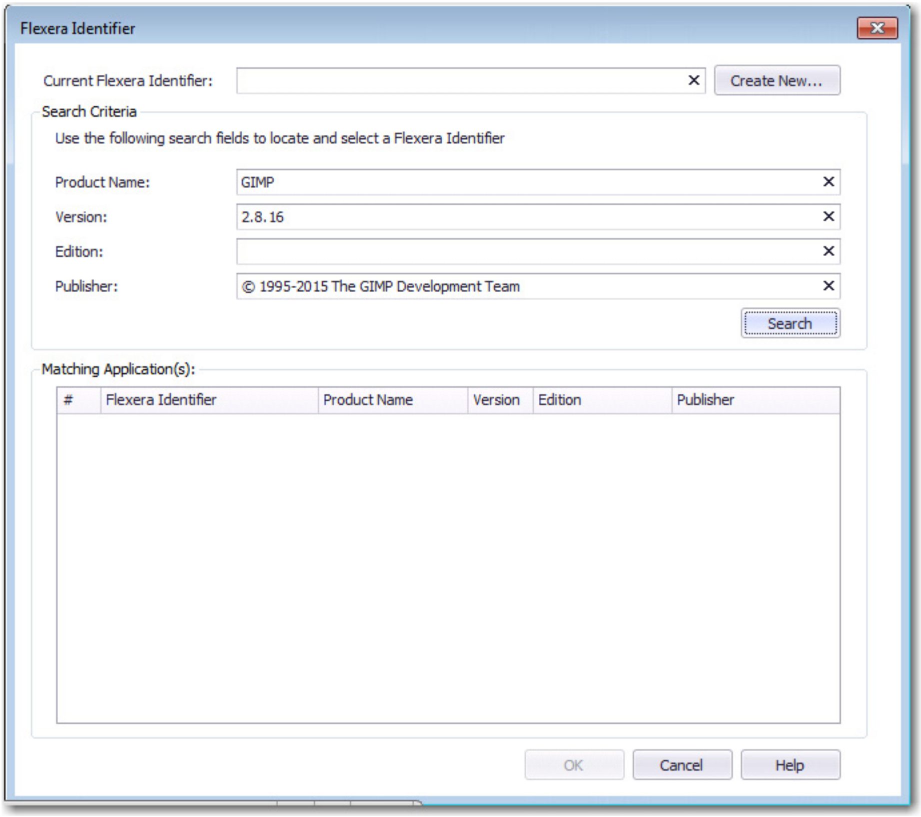


The **Application Search Results** dialog box opens, listing all applications in the Application Catalog that do not have an associated Flexera Identifier.



Product Name	Version	Publisher
iMovie	1768	
Install	9.0.0.115	
Adobe Reader	11.0.10	Copyright 1984-2012 Adobe System...
iTunes12.4.1		
GIMP	2.8.16	© 1995-2015 The GIMP Developmen...
mysql-5.7.13-osx10.11-x86_64		
Office Setup Assistant	14.2.0	© 2010 Microsoft Corporation. All ri...
AdobeAIR_3.4	3.4.0.2540	Copyright © 2007-2012 Adobe Syst...
Autodesk SketchBook	8.1.1	Autodesk
FileZilla_3.2.4.1_win32-setup	3.2.4.1	
TigerText Secure Messaging for Business	6.1.1	TigerText, Inc.
QuickBooks	2.4.5	Intuit Inc.

2. Select an application in the list and click **Assign Flexera ID**. The **Flexera Identifier** dialog box opens.



Flexera Identifier

Current Flexera Identifier:

Search Criteria

Use the following search fields to locate and select a Flexera Identifier

Product Name: GIMP

Version: 2.8.16

Edition:

Publisher: © 1995-2015 The GIMP Development Team

Matching Application(s):

#	Flexera Identifier	Product Name	Version	Edition	Publisher

3. Edit the text in the **Search Criteria** fields to either correct the information or make it less specific, and then click **Search**. A list of possible matching applications will be generated and will be listed in the **Matching Application(s)** list.

4. Do one of the following:

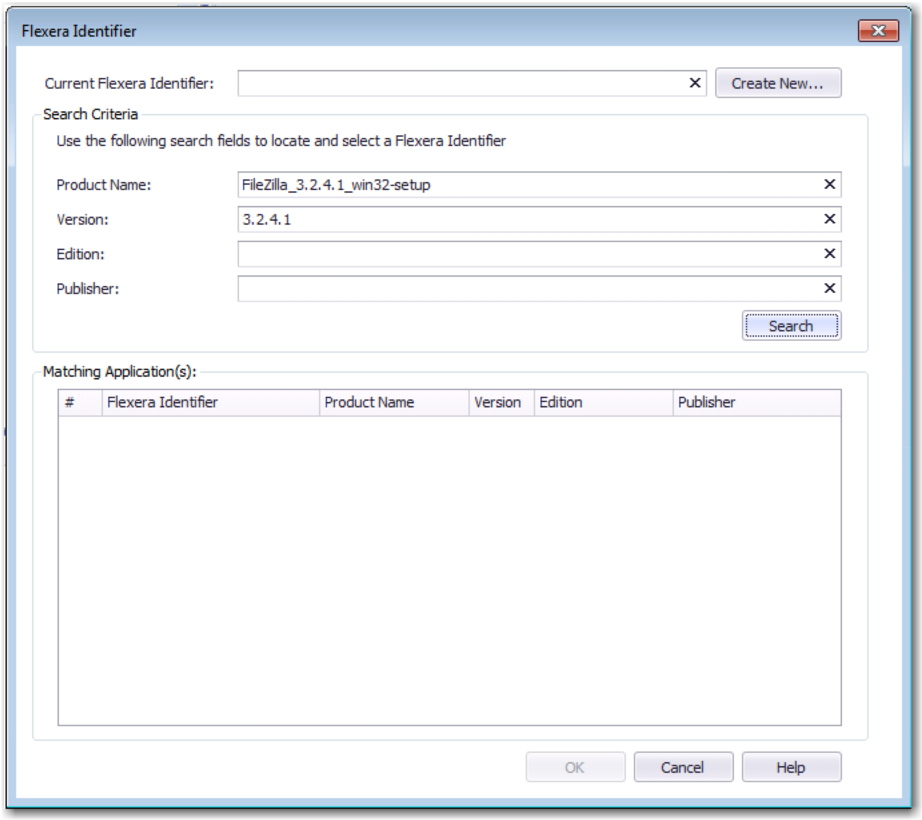
   - **Matching application found**—If the correct matching application is listed, select it from the list and click **OK**. The selected Flexera Identifier will be saved in the Application Catalog and will be listed on the **General Information** tab of the **Application View** for that application.
   - **Matching application not found**—If a matching application is not listed, continue with the steps in Ability to Create Local Flexera Identifier Entries for Internal or Repackaged Applications to create a new local Flexera Identifier.

Ability to Create Local Flexera Identifier Entries for Internal or Repackaged Applications

In AdminStudio 2016, you can create new local Flexera Identifier entries for the FlexNet Manager Suite Application Recognition Library. These may be required for internally developed applications, repackaged applications, and other applications that are not recognized by FlexNet Manager Suite.

Task ***To create a local Flexera Identifier for an application:***

1. Open the **Application View** of the unrecognized application and click the browse button in the empty **Flexera Identifier** field. The **Flexera Identifier** dialog box opens.

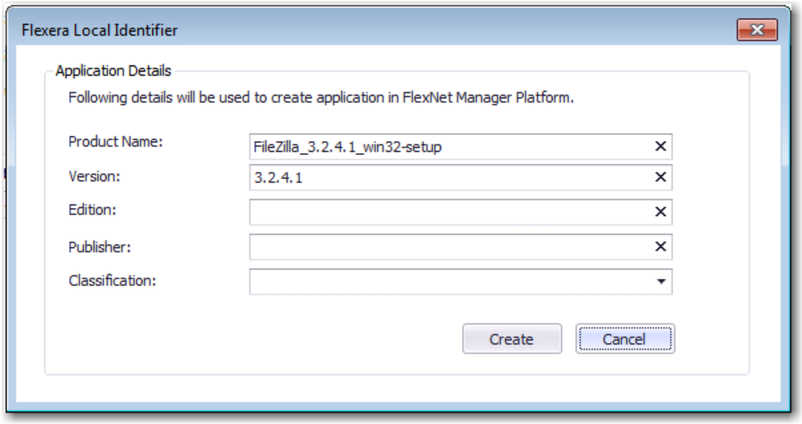


Product Name: FileZilla_3.2.4.1_win32-setup

Version: 3.2.4.1

2. Use the search fields to locate and select a Flexera Identifier.

3. If no Flexera Identifier is found, click **Create New**. The **Flexera Local Identifier** dialog box opens.



Flexera Local Identifier

Application Details

Following details will be used to create application in FlexNet Manager Platform.

Product Name: FileZilla_3.2.4.1_win32-setup

Version: 3.2.4.1

Edition:

Publisher:

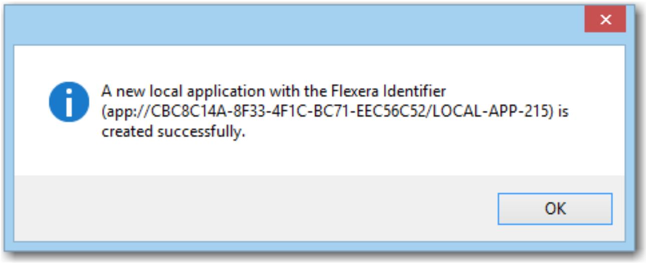
Classification:

4. Enter the following information:

Property	Description
Product Name	The basic name of the application, excluding references to versions or editions, and without mentioning the publisher.
Version	The release number (or release identifier) of an application.
Edition	Enter the edition of this application.
Publisher	The name of the publisher of this software, responsible for its development and distribution.

Property	Description
Classification	To indicate how this application is classified, select one of the following options: ● **Beta**—A pre-release application (covers such items as beta releases, alpha releases, or release candidates) that you have under some special arrangement. ● **Commercial**—The application requires a license to be purchased for use in a commercial setting. ● **Freeware**—Licensed for use in a commercial environment free-of-charge. ● **Malware**—A potentially harmful application (a virus, Trojan, and the like), and should be treated as malware. If installations of this application are identified, you need to address the corresponding incidents or security issues. ● **Shareware**—The application is available for downloading from web sites, and typically uses a “try-before-you-buy” licensing model that might include reminder messages, functional limitations, or other restrictions until a full license is purchased. ● **X Rated**—The application contains potentially objectionable or sexually explicit material. You might want to consider whether corporate policies require any action. ● **Update**—The application represents an update, for example, a service pack, to another application, and is issued for free to all customers regardless of purchasing agreements or support contracts (a “minor” update).

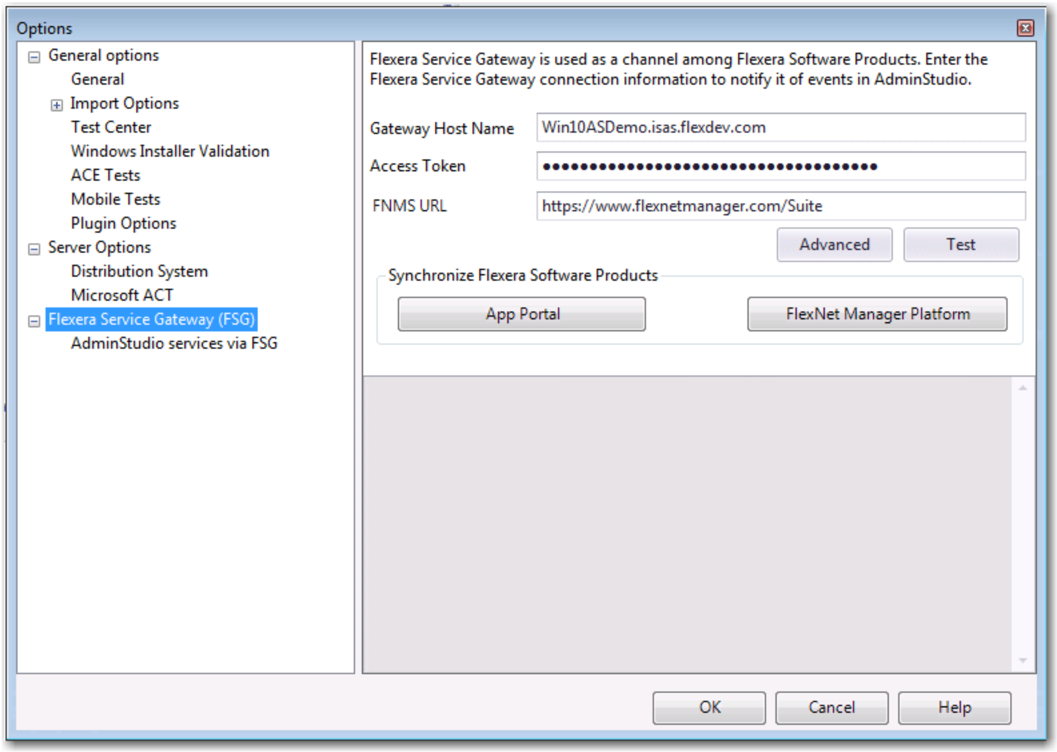
5. Click **Create**. A confirmation message appears stating that a new local Flexera Identifier has been created.



A new local application with the Flexera Identifier (app://CBC8C14A-8F33-4F1C-BC71-EEC56C52/LOCAL-APP-215) is created successfully.

Ability to Enter FlexNet Manager Suite URL

In AdminStudio 2016, you can now more easily configure your connection to FlexNet Manager Suite by entering the FlexNet Manager Suite URL in the **FNMS URL** field on the **Flexera Service Gateway (FSG)** tab of the Application Manager **Options** dialog box.



Flexera Service Gateway is used as a channel among Flexera Software Products. Enter the Flexera Service Gateway connection information to notify it of events in AdminStudio.

Gateway Host Name: Win10ASDemo.isas.flexdev.com
Access Token: ••••••••••••••••••••••••••••••••••••
FNMS URL: https://www.flexnetmanager.com/Suite

Figure 51: FNMS URL Field on Flexera Service Gateway (FSG) Tab of Options Dialog Box

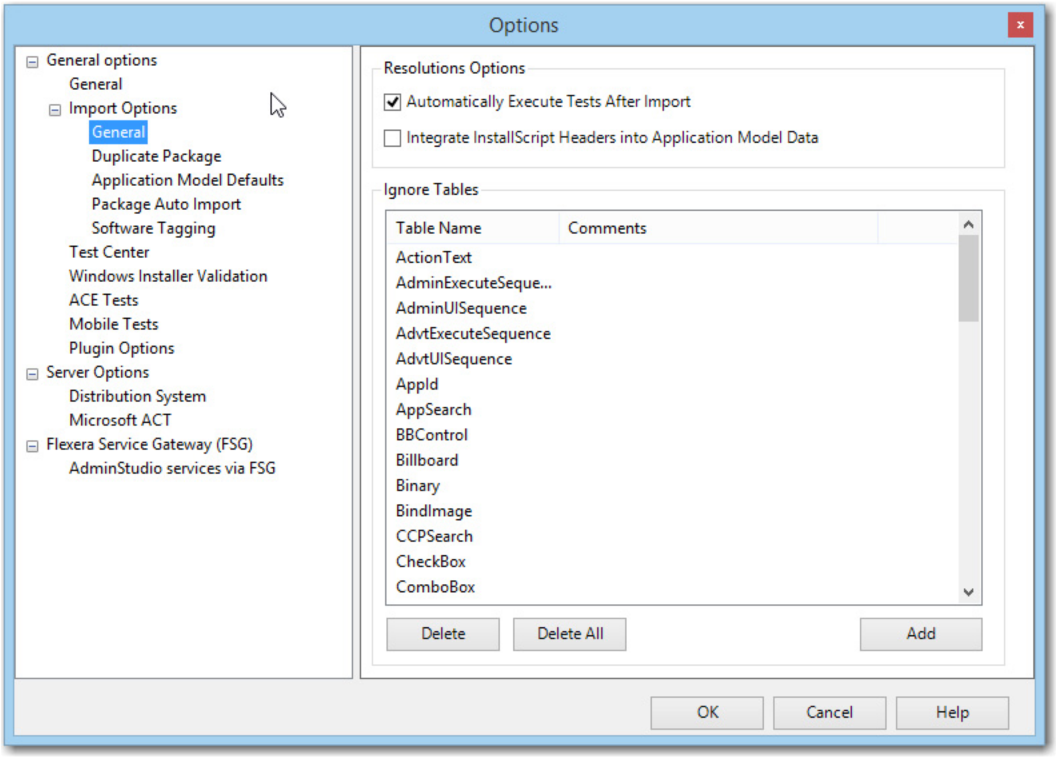
Enhanced Automation Capabilities to Streamline Application Import and Testing

AdminStudio 2016 offers several new automation features that help you to streamline the application import and testing process.

- Applications Are Automatically Tested Upon Import
- Enhancements to Package Auto Import Feature
- Enhanced Control When Using Import Wizard to Perform Batch Import from a Directory

Applications Are Automatically Tested Upon Import

In AdminStudio 2016, Application Manager is now configured to automatically test packages against all selected compatibility, best practices, and risk assessment tests as part of the import process. By default, the **Automatically Execute Tests After Import** option on the **Import Options** tab of the Application Manager **Options** dialog box is selected. In previous releases, this option was not selected by default.



Resolutions Options

☑ Automatically Execute Tests After Import

□ Integrate InstallScript Headers into Application Model Data

Ignore Tables

Table Name	Comments
ActionText	
AdminExecuteSeque...	
AdminUISequence	
AdvtExecuteSequence	
AdvtUISequence	
AppId	
AppSearch	
BBControl	
Billboard	
Binary	
BindImage	
CCPSearch	
CheckBox	
ComboBox	

Figure 52: Automatically Execute Tests After Import Option

While having this option selected will mean longer import times for each application, packages will have all testing details populated immediately after import. However, if you are concerned with the length of import time, you may want to clear the selection of this option.

***Note •** In previous releases of AdminStudio, the **Automatically Execute Tests After Import** option was not selected by default; in AdminStudio 2016, this option is selected by default. Therefore, for users upgrading from previous releases of AdminStudio, they will inherit the new default selection, which means that, by default, this option will be selected and tests will be run immediately after import. To have the same user experience as you did in previous versions, you need to clear the selection of this option.*

Enhancements to Package Auto Import Feature

The Application Manager **Package Auto Import** feature enables you to automatically import or re-import packages in shared network directories into your Application Catalog. Application Manager periodically checks these directories and imports all new packages it finds, and reimports all updated packages.

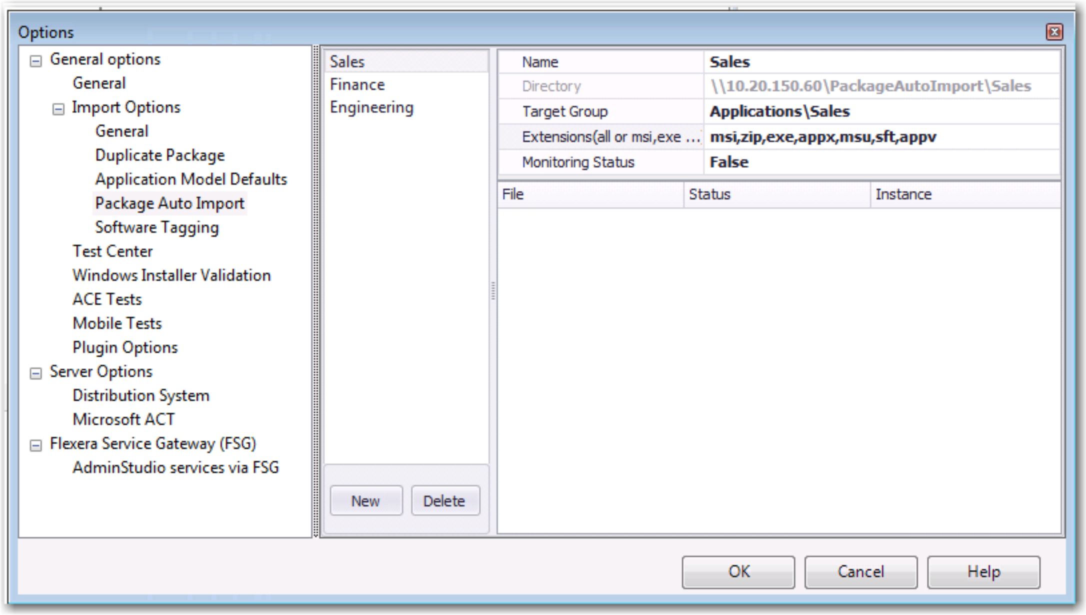
In AdminStudio 2016, the following enhancements have been added to this feature to make it even more powerful and useful:

- Can Monitor Multiple Directories Simultaneously
- Can Select File Types to Monitor
- Can Perform Batch Auto Import of Microsoft Security Patch Files and iOS Enterprise Policy Files

Can Monitor Multiple Directories Simultaneously

In previous releases, when setting up the Package Auto Import feature, you could only specify a single directory to scan for new or updated packages to import.

In AdminStudio 2016, you can specify multiple directories to monitor. On the **General Options > Import Options > Package Auto Import** tab of the **Options** dialog box, you can add a directory to monitor by clicking the **New** button. There is no limit to the number of directories you can monitor.



Sales
Finance
Engineering

Name	Sales
Directory	\\10.20.150.60\PackageAutoImport\Sales
Target Group	Applications\Sales
Extensions(all or msi,exe ...	msi,zip,exe,appx,msu,sft,appv
Monitoring Status	False

Figure 53: Import Options > Package Auto Import Tab of Options Dialog Box

After the initial auto-import operation, packages will be imported or reimported into the Application Catalog whenever:

- A new package is added to one of the monitored directories.
- An existing package in one of the monitored directories is updated.

Each time you specify a directory to monitor, you need to provide the following information.

Property	Description
Directory	Enter the directory to monitor. You can either enter a directory path or click the browse button and select a directory. ***Important*** *• The directory must be in UNC format and it must be a shared directory.*
Target Group	Specify the group in the Application Manager tree into which the packages will be imported.
Monitoring Status	Set this option to **True** to begin monitoring this directory for new and updated packages. Set it to **False** to stop monitoring the directory.
Extensions	Click the browse button to open the **Select Watcher Extensions** dialog box, where you will specify the package types to import, as described in Can Select File Types to Monitor.

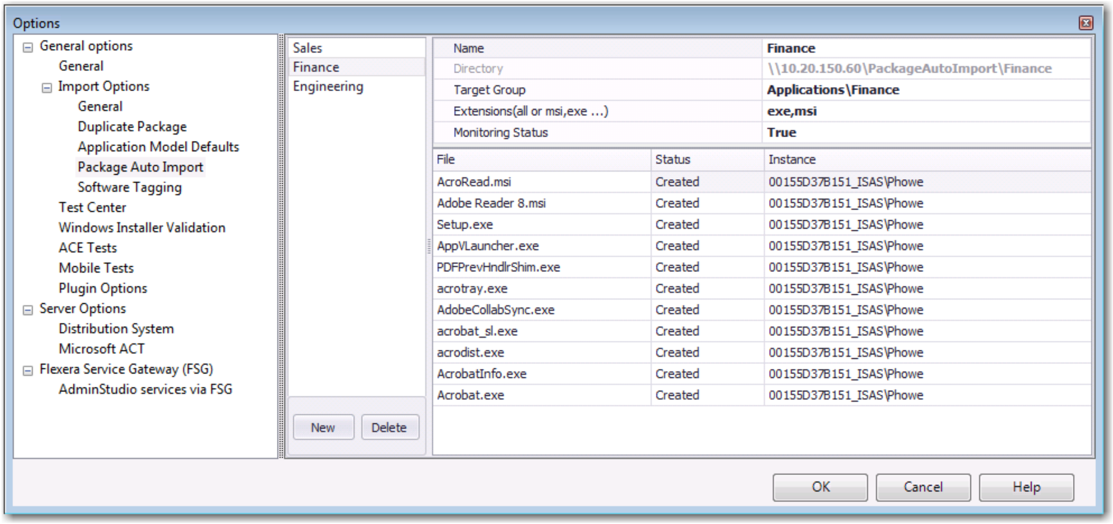
The monitoring of the specified Package Auto Import directories is a task that can be distributed among the AdminStudio Host processes of all of the installations of AdminStudio in your network that have access to the specified shared directory.

After you have specified a directory on the **Package Auto Import** tab and have set the **Monitoring Status** field to **True**, the AdminStudio Host processes that are connected to the same Application Catalog will begin to monitor these directories.

Packages in the monitored directories will be imported as soon as one of the AdminStudio Host processes connected to the same Application Catalog is running but is not currently being used. This ensures that if someone is actively using one of AdminStudio’s tools, the import (and subsequent testing) of packages by the Package Auto Import process will not slow down the performance of AdminStudio.

Tip *• To make sure that the batch import activities of the Package Auto Import process take place when AdminStudio tools are not in use, you could launch AdminStudioHost.exe as a scheduled task during off peak hours and it will perform the import of packages from the monitored directories at that time.*

After a directory’s **Monitoring Status** has been set to **True**, the packages (of the selected package type) in that directory are listed on the **Package Auto Import** tab.



Name	**Finance**
Directory	\\10.20.150.60\PackageAutoImport\Finance
Target Group	**Applications\Finance**
Extensions(all or msi,exe ...)	**exe,msi**
Monitoring Status	**True**

File	Status	Instance
AcroRead.msi	Created	00155D37B151_ISAS\Phowe
Adobe Reader 8.msi	Created	00155D37B151_ISAS\Phowe
Setup.exe	Created	00155D37B151_ISAS\Phowe
AppVLauncher.exe	Created	00155D37B151_ISAS\Phowe
PDFPrevHndlrShim.exe	Created	00155D37B151_ISAS\Phowe
acrotray.exe	Created	00155D37B151_ISAS\Phowe
AdobeCollabSync.exe	Created	00155D37B151_ISAS\Phowe
acrobat_sl.exe	Created	00155D37B151_ISAS\Phowe
acrodist.exe	Created	00155D37B151_ISAS\Phowe
AcrobatInfo.exe	Created	00155D37B151_ISAS\Phowe
Acrobat.exe	Created	00155D37B151_ISAS\Phowe

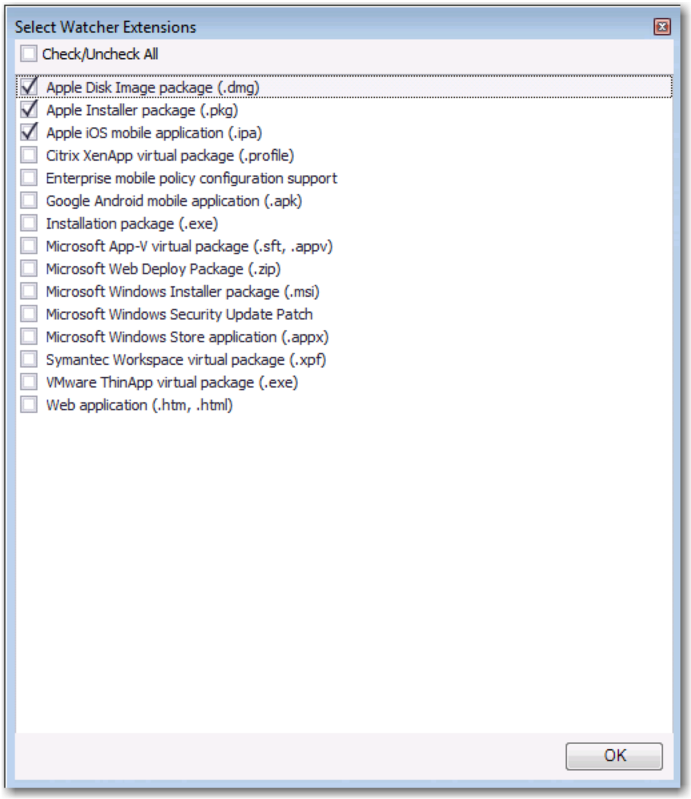
Figure 54: Monitoring the Import of Packages in a Directory on the Package Auto Import Tab

In the **Status** column, packages will have one of the following statuses:

- **Created**—Package will be imported as soon as one of the AdminStudio Host processes connected to this Application Catalog is running but is not currently being used.
- **Imported**—Package has been imported into the Application Catalog.
- **Canceled**—User canceled the import of the package into the Application Catalog.
- **Error**—There was an error during import of the package into the Application Catalog.

Can Select File Types to Monitor

When setting up a directory to monitor for a Package Auto Import process, you can now specify the specific package types in that directory to import. To do this, you click the browse button in the **Extensions** field of the **Package Auto Import** tab of the **Options** dialog box to open the **Select Watcher Extensions** dialog box.



□ Check/Uncheck All

☑ Apple Disk Image package (.dmg)
☑ Apple Installer package (.pkg)
☑ Apple iOS mobile application (.ipa)
□ Citrix XenApp virtual package (.profile)
□ Enterprise mobile policy configuration support
□ Google Android mobile application (.apk)
□ Installation package (.exe)
□ Microsoft App-V virtual package (.sft, .appv)
□ Microsoft Web Deploy Package (.zip)
□ Microsoft Windows Installer package (.msi)
□ Microsoft Windows Security Update Patch
□ Microsoft Windows Store application (.appx)
□ Symantec Workspace virtual package (.xpf)
□ VMware ThinApp virtual package (.exe)
□ Web application (.htm, .html)

Figure 55: Select Watcher Extensions Dialog Box

Only those packages of the package type you select on this dialog box will be imported from the monitored directory.

Can Perform Batch Auto Import of Microsoft Security Patch Files and iOS Enterprise Policy Files

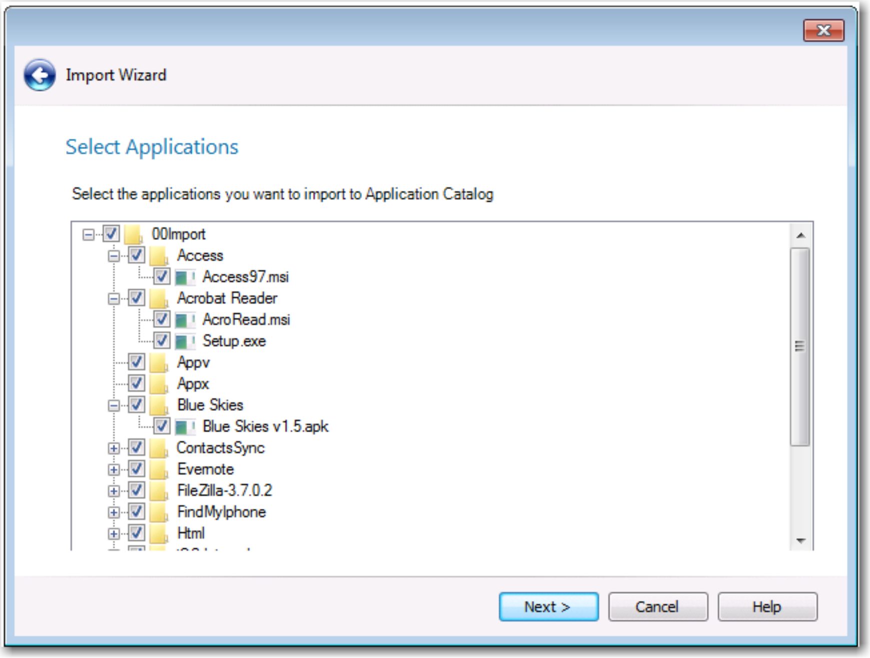
Previously, you could only import Microsoft Security Patch files (`.msu`) and iOS Enterprise Policy Configuration Files (`.mobileconfig` or `.plist`) one at a time using the Import Wizard.

In AdminStudio 2016, you can now perform a batch import of Microsoft Security Patch files and iOS Enterprise Policy Configuration files using the Package Auto Import feature. To do this, select the **Microsoft Windows Security Update Patch** and/or the **Enterprise mobile policy configuration support** options on the **Select Watcher Extensions** dialog box.

Enhanced Control When Using Import Wizard to Perform Batch Import from a Directory

When using the Import Wizard to import a folder of applications, you can now choose to import only the packages that you select rather than all packages of the selected package type in the directory.

After you select the package types that you want to import from the selected directory on the **Package Type Selection** panel of the Import Wizard, a new panel named **Select Applications** opens, prompting you to select specific applications to import from the selected directory.



Import Wizard

Select Applications

Select the applications you want to import to Application Catalog

- ☑ 00Import
  - ☑ Access
    - ☑ Access97.msi
  - ☑ Acrobat Reader
    - ☑ AcroRead.msi
    - ☑ Setup.exe
  - ☑ Appv
  - ☑ Appx
  - ☑ Blue Skies
    - ☑ Blue Skies v1.5.apk
  - ☑ ContactsSync
  - ☑ Evernote
  - ☑ FileZilla-3.7.0.2
  - ☑ FindMyIphone
  - ☑ Html

Figure 56: Select Applications Panel of Import Wizard

By default, all packages of the selected package type are selected. You can clear the selection of any packages you do not want to import.

Automated Application Converter Enhancements

AdminStudio 2016 contains several enhancements to the Automated Application Converter tool to expand its functionality.

- Automated Application Converter for Repackaging Now Included in Professional and Enterprise Editions
- Ability to Edit Default Automated Application Converter Settings From Application Manager
- Can Edit Conversion Settings From the Conversion Wizard on a Per-Run Basis

Automated Application Converter for Repackaging Now Included in Professional and Enterprise Editions

In AdminStudio 2016, the Professional and Enterprise editions now include Automated Application Converter—which you can use to perform automated repackaging on a virtual machine—without requiring you to purchase the Virtualization add-on pack.

- **Professional edition**—You can perform automated repackaging one package at a time.
- **Enterprise edition**—You can perform batch automated repackaging of a group of packages.

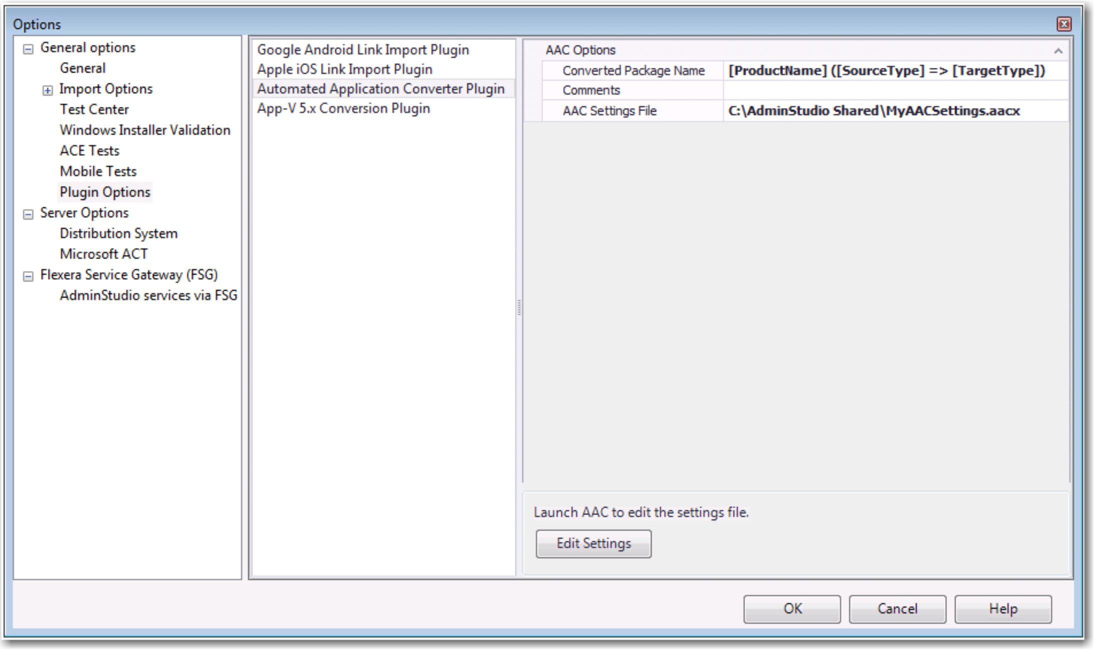
Note • *In order to use Automated Application Converter to convert packages to virtual formats, you will still be required to purchase the Virtualization add-on pack.*

You can choose to perform automated repackaging using either the Automated Application Converter interface or by using the Conversion Wizard, which is launched from Application Manager by right-clicking on a package or group of packages and then selecting **Launch Conversion Wizard** from the shortcut menu. Previously, the Conversion Wizard feature of Application Manager was only included with AdminStudio if you purchased the Virtualization add-on pack.

Ability to Edit Default Automated Application Converter Settings From Application Manager

In previous releases, you were able to use the Application Manager Conversion Wizard to quickly convert one or multiple Windows Installer packages or legacy installers to virtual packages (of the specified type) using *default* Automated Application Converter settings. However, to edit those default settings, you were required to launch Automated Application Converter and open the default project file.

In AdminStudio 2016, you can now edit the default settings file by clicking the new **Edit Settings** button on the **Plugin Options > Automated Application Converter Plugin** tab of the **Options** dialog box:

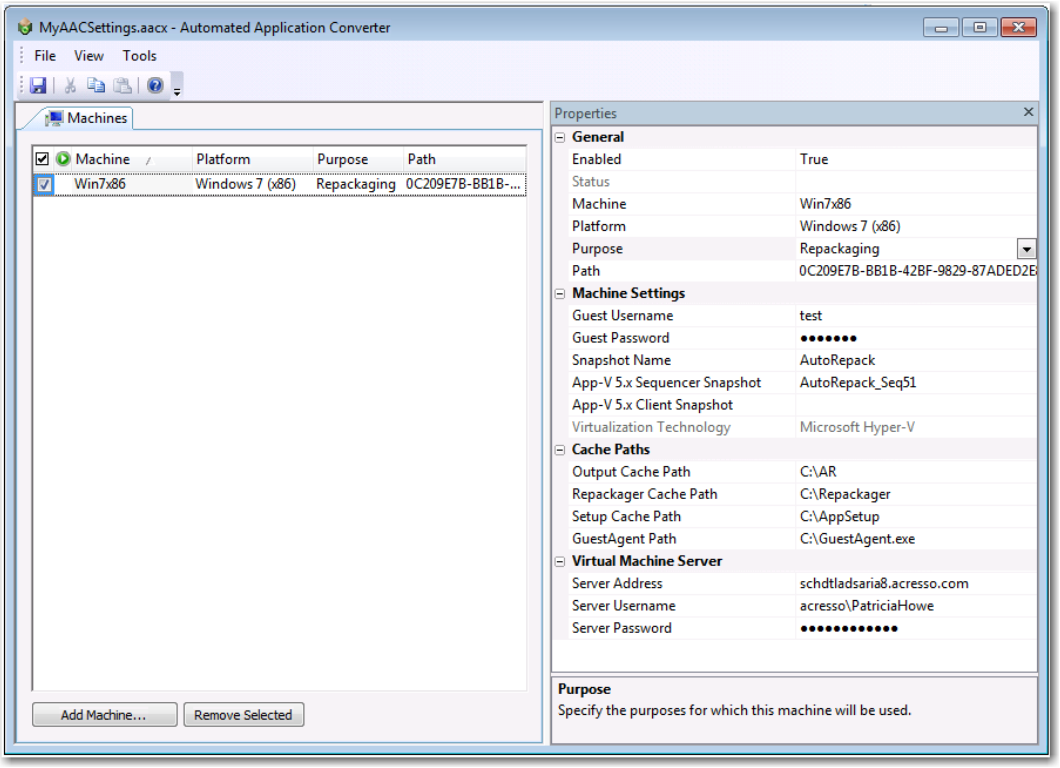

Options
General options
General
Import Options
Test Center
Windows Installer Validation
ACE Tests
Mobile Tests
Plugin Options
Server Options
Distribution System
Microsoft ACT
Flexera Service Gateway (FSG)
AdminStudio services via FSG
Google Android Link Import Plugin
Apple iOS Link Import Plugin
Automated Application Converter Plugin
App-V 5.x Conversion Plugin
AAC Options
Edit Settings
OK
Cancel
Help

Converted Package Name	[ProductName] ([SourceType] => [TargetType])
Comments	
AAC Settings File	C:\AdminStudio Shared\MyAACSettings.aacx

Launch AAC to edit the settings file.

Figure 57: Edit Settings Button on Plugin Options > Automated Application Converter Plugin Tab

When you click **Edit Settings**, a limited version of Automated Application Converter opens, displaying the **Machines** tab, where you can quickly edit any of these default conversion settings and save them into this default project file.



☑ Machine	Platform	Purpose	Path
☑ Win7x86	Windows 7 (x86)	Repackaging	0C209E7B-BB1B-...

General	
Enabled	True
Status	
Machine	Win7x86
Platform	Windows 7 (x86)
Purpose	Repackaging
Path	0C209E7B-BB1B-42BF-9829-87ADED2E
Machine Settings	
Guest Username	test
Guest Password	●●●●●●●
Snapshot Name	AutoRepack
App-V 5.x Sequencer Snapshot	AutoRepack_Seq51
App-V 5.x Client Snapshot	
Virtualization Technology	Microsoft Hyper-V
Cache Paths	
Output Cache Path	C:\AR
Repackager Cache Path	C:\Repackager
Setup Cache Path	C:\AppSetup
GuestAgent Path	C:\GuestAgent.exe
Virtual Machine Server	
Server Address	schdtladsaria8.acresso.com
Server Username	acresso\PatriciaHowe
Server Password	●●●●●●●●●●●●

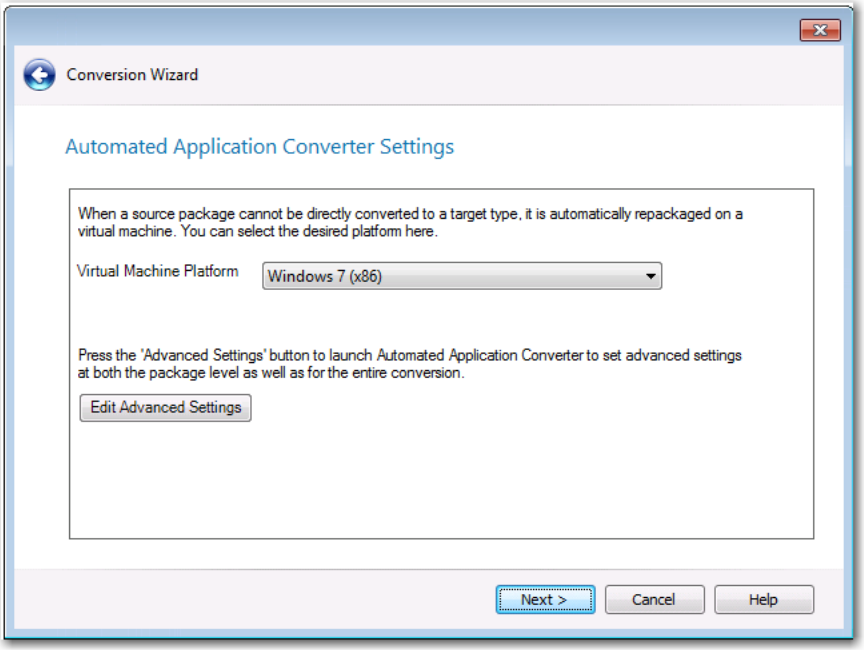
Purpose
Specify the purposes for which this machine will be used.

Figure 58: Editing the Default Automated Application Converter Settings File

Can Edit Conversion Settings From the Conversion Wizard on a Per-Run Basis

In AdminStudio 2016, you now have to option of modifying the default conversion settings for a single run of the Conversion Wizard.

There is a new panel in the Conversion Wizard entitled **Automated Application Converter Settings**. On this panel, the virtual machine platforms defined in the settings file are listed in the **Virtual Machine Platform** list, and you can choose which platform to use for this run of the Conversion Wizard.



Conversion Wizard

Automated Application Converter Settings

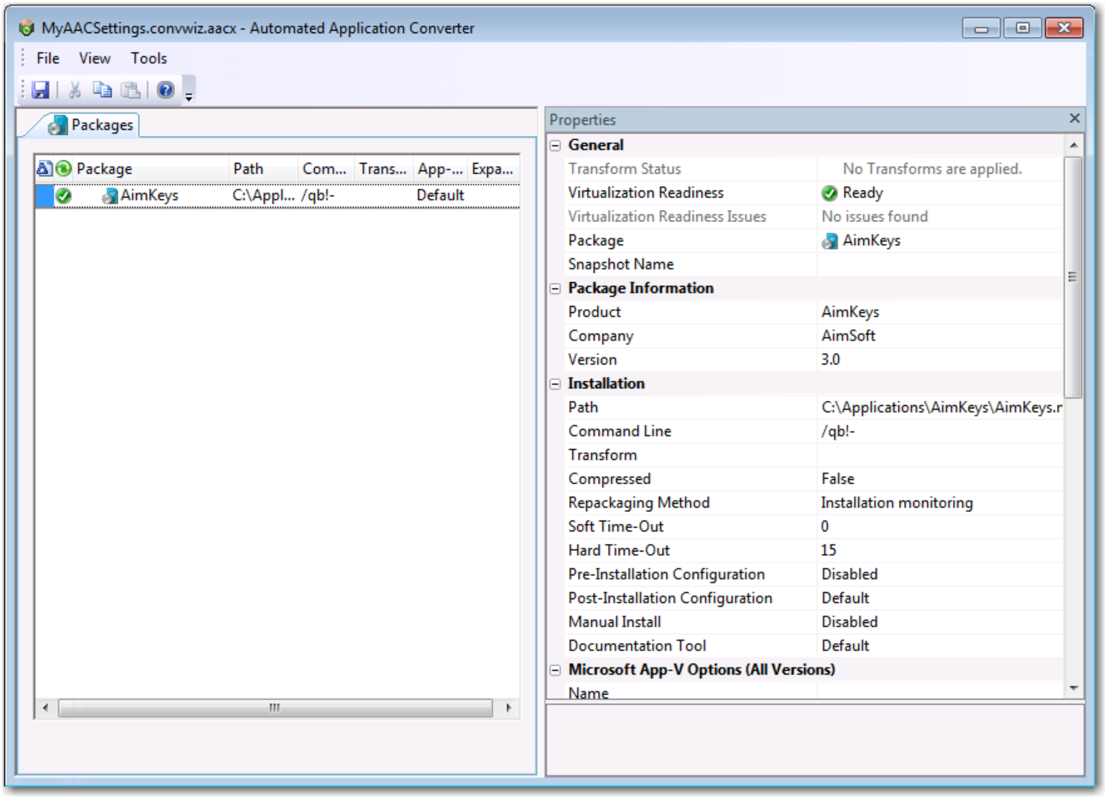
When a source package cannot be directly converted to a target type, it is automatically repackaged on a virtual machine. You can select the desired platform here.

Virtual Machine Platform: Windows 7 (x86)

Press the 'Advanced Settings' button to launch Automated Application Converter to set advanced settings at both the package level as well as for the entire conversion.

Figure 59: Automated Application Converter Settings Panel of Conversion Wizard

If you want to edit additional advanced settings, you can click the **Edit Advanced Settings** button. A copy of the default conversion settings file (that is specified on the **Plugin Options > Automated Application Converter Plugin** tab of the **Options** dialog box) is opened, displaying the **Packages** tab of Automated Application Converter.


MyAACSettings.convwiz.aacx - Automated Application Converter
File
View
Tools
Packages
Properties

Package	Path	Com...	Trans...	App-...	Expa...
AimKeys	C:\Appl...	/qb!-		Default	

General	
Transform Status	No Transforms are applied.
Virtualization Readiness	✓ Ready
Virtualization Readiness Issues	No issues found
Package	AimKeys
Snapshot Name	
Package Information	
Product	AimKeys
Company	AimSoft
Version	3.0
Installation	
Path	C:\Applications\AimKeys\AimKeys.r
Command Line	/qb!-
Transform	
Compressed	False
Repackaging Method	Installation monitoring
Soft Time-Out	0
Hard Time-Out	15
Pre-Installation Configuration	Disabled
Post-Installation Configuration	Default
Manual Install	Disabled
Documentation Tool	Default
Microsoft App-V Options (All Versions)	
Name	

Figure 60: Editing the Default Conversion Settings

You can change any of these settings. When you save this file and exit the **Packages** tab of Automated Application Converter, you are returned to the Conversion Wizard where you can proceed with the conversion.

New Wizard to Install and Launch Windows Installer Packages on a Virtual Machine for Testing

In AdminStudio 2016, the process of resetting a VM snapshot, powering it on, and copying the application for testing into it can be automated using the new **Test on Virtual Machine Wizard**, which is available with Professional and Enterprise Editions.

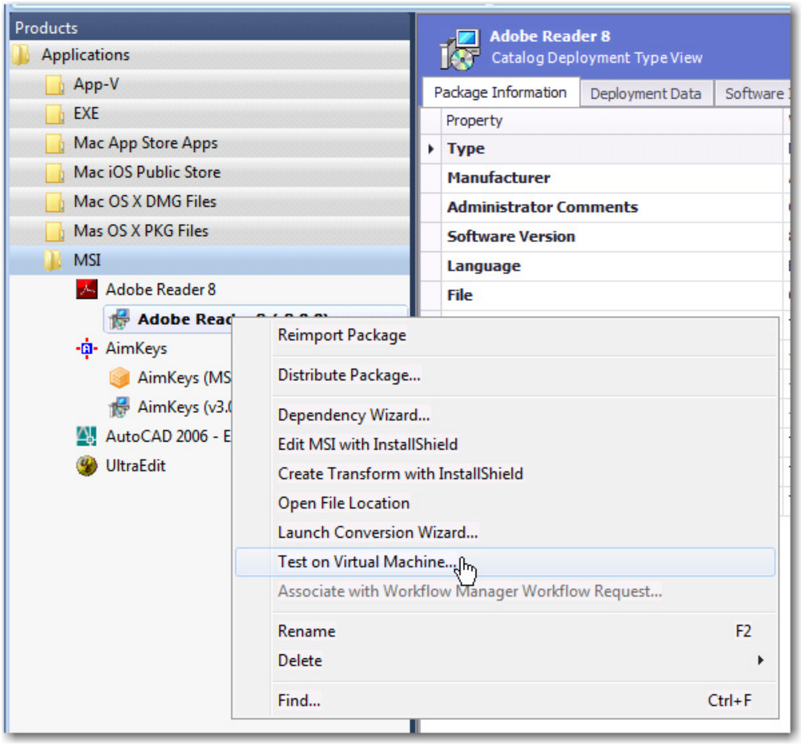
The **Test on Virtual Machine Wizard** uses the capability of the Automated Application Converter tool to quickly launch a specified virtual machine, install a selected Windows Installer (`.msi`) or installation executable (`.exe`) package, and enable you to launch a remote desktop session to that virtual machine so that you can perform testing.

***Note** • Both legacy installers and complex installer executables (which contain bundled Windows Installer packages) can be tested using the Test on Virtual Machine Wizard.*

To use the Test on Virtual Machine Wizard to test a package, perform the following steps.

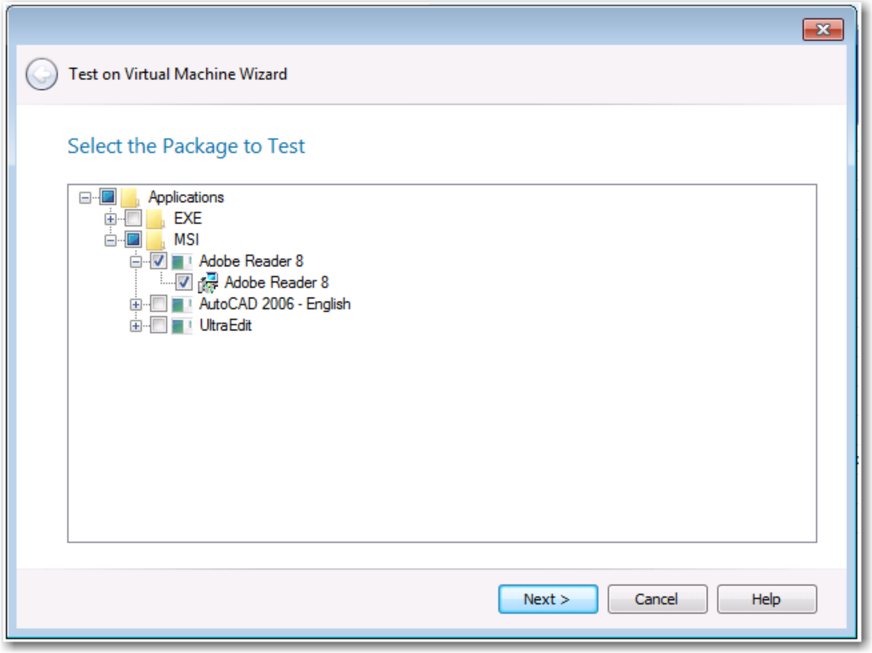
Task ***To use the Test on Virtual Machine Wizard to test a package:***

1. Right click a Windows Installer (`.msi`) or installer executable (`.exe`) package (or on an application containing an `.msi` or `.exe` package) and select **Test on Virtual Machine** from the shortcut menu.



***Note** • The **Test on Virtual Machine** selection on the shortcut menu is available on both the **Catalog** and the **Test Center** tabs of Application Manager.*

The **Select Package to Test** panel opens. The Application Manager tree is displayed, with the package that was selected when you opened the wizard automatically selected.



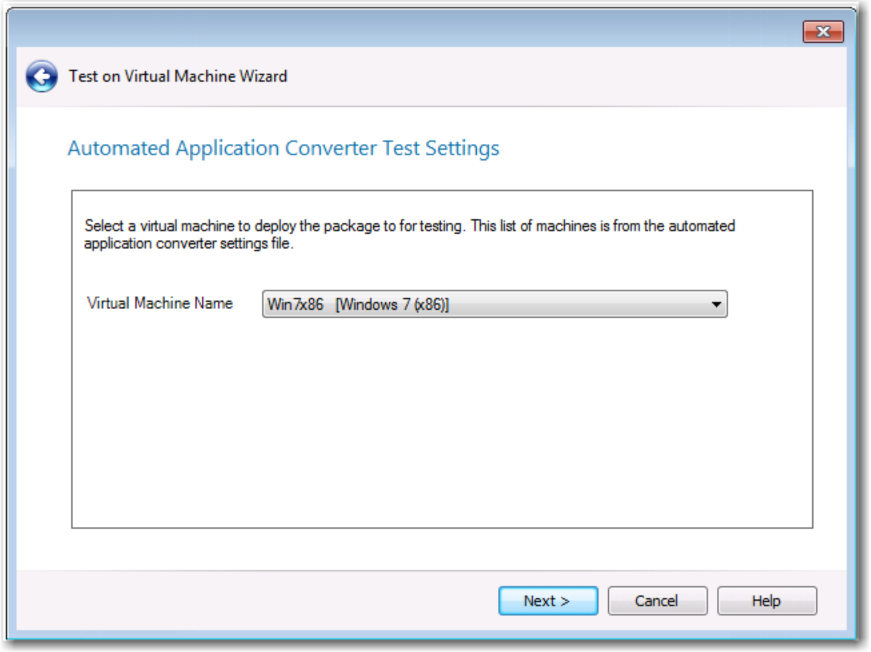
Test on Virtual Machine Wizard

Select the Package to Test

- ■ Applications
  - ☐ EXE
  - ■ MSI
    - ☑ Adobe Reader 8
      - ☑ Adobe Reader 8
    - ☐ AutoCAD 2006 - English
    - ☐ UltraEdit

2. Confirm the selection of the package that you want to test and click **Next**.

   The **Automated Application Converter Test Settings** panel opens, listing the virtual machines defined in the Automated Application Converter settings file that is selected on the **Plugin Options > Automated Application Converter Plugin** tab of the **Options** dialog box:



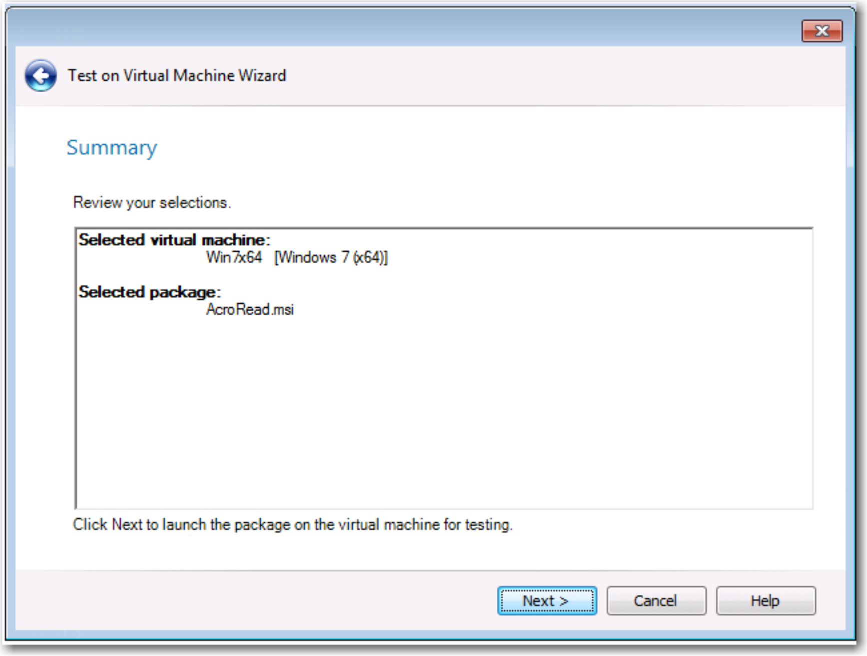
Test on Virtual Machine Wizard

Automated Application Converter Test Settings

Select a virtual machine to deploy the package to for testing. This list of machines is from the automated application converter settings file.

Virtual Machine Name: Win7x86 [Windows 7 (x86)]

3. Select the name of the virtual machine that you want to use for testing and click **Next**. The **Summary** panel opens.



Summary

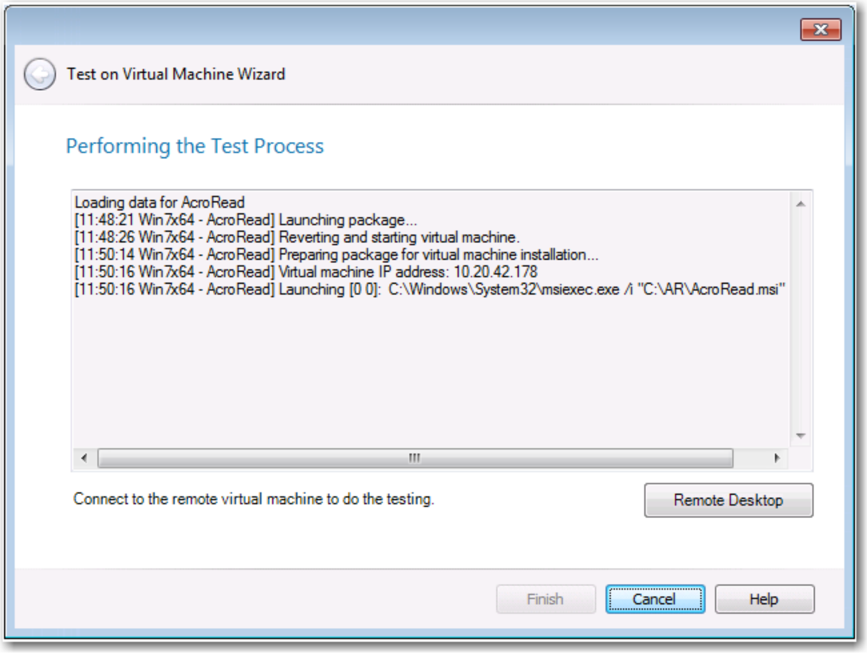
Review your selections.

Selected virtual machine:
Win7x64 [Windows 7 (x64)]

Selected package:
AcroRead.msi

Click Next to launch the package on the virtual machine for testing.

4. Click **Next** to launch the package on the specified virtual machine for testing. The **Performing the Test Process** panel opens, listing progress messages. When the package has been installed and launched on the virtual machine, the **Remote Desktop** button will become enabled.

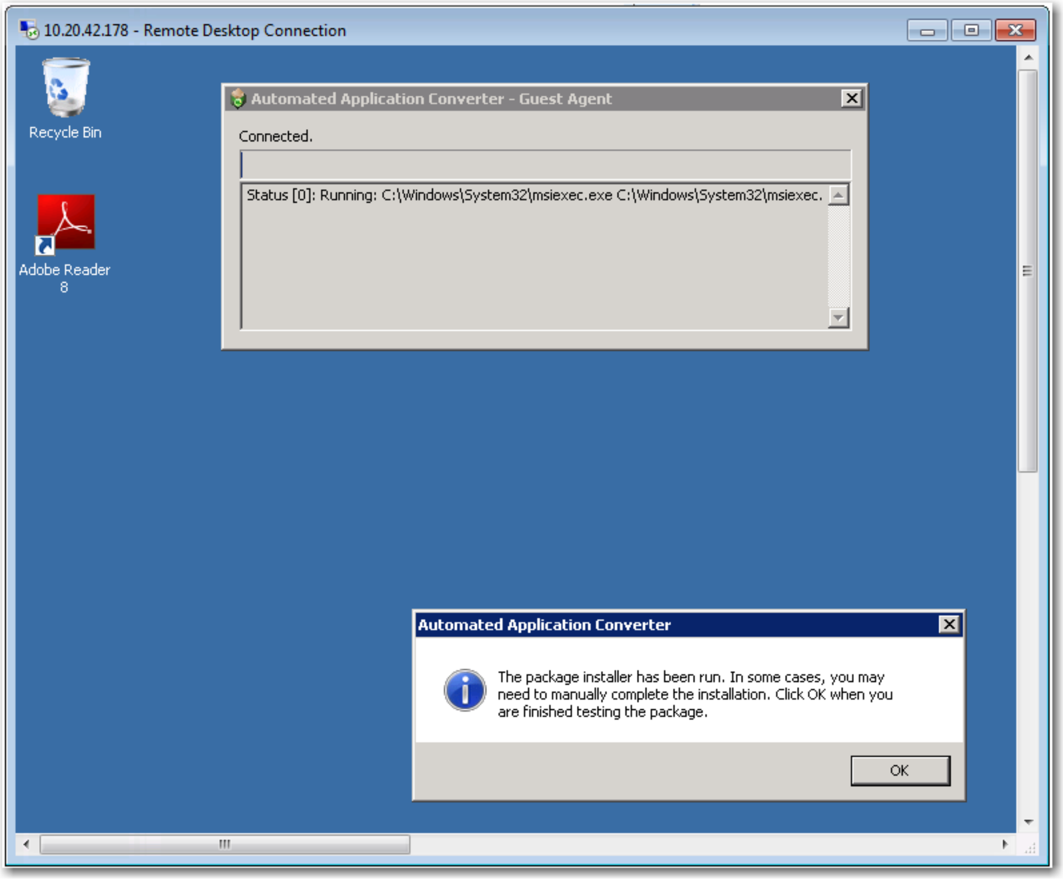


Performing the Test Process

```
Loading data for AcroRead
[11:48:21 Win7x64 - AcroRead] Launching package...
[11:48:26 Win7x64 - AcroRead] Reverting and starting virtual machine.
[11:50:14 Win7x64 - AcroRead] Preparing package for virtual machine installation...
[11:50:16 Win7x64 - AcroRead] Virtual machine IP address: 10.20.42.178
[11:50:16 Win7x64 - AcroRead] Launching [0 0]:  C:\Windows\System32\msiexec.exe /i "C:\AR\AcroRead.msi"
```

Connect to the remote virtual machine to do the testing.

5. Click **Remote Desktop** to connect to the virtual machine and perform testing. You may be prompted for login credentials to the virtual machine image. A Remote Desktop session opens displaying the virtual image where this package has been installed.


10.20.42.178 - Remote Desktop Connection
Recycle Bin
Adobe Reader 8
Automated Application Converter - Guest Agent
Automated Application Converter
OK

Connected.

Status [0]: Running: C:\Windows\System32\msiexec.exe C:\Windows\System32\msiexec.

The package installer has been run. In some cases, you may need to manually complete the installation. Click OK when you are finished testing the package.

6. Use the installed shortcuts to launch the package and perform the desired testing.

7. When you have finished testing the package, click **OK** to close the Remote Desktop session and shut down the virtual machine.

8. Return to the **Test on Virtual Machine Wizard** and click **Finish** to close the wizard.

Productivity Shortcuts

Several productivity shortcuts have been added to AdminStudio in the 2016 release.

- New Shortcut to Automatically Create Transform for Selected Package in InstallShield
- Ability to Browse to Package Location from Application Manager Tree

New Shortcut to Automatically Create Transform for Selected Package in InstallShield

In previous releases, you were able to select **Edit with InstallShield** from the shortcut menu to quickly open a Windows Installer package in InstallShield. However, the only option you had was to open the package in Direct Edit Mode, which results in modifying the Windows Installer `.msi` file itself.

In AdminStudio 2016, you can quickly create a new transform project for a Windows Installer package by right-clicking on the package in the Application Manager tree and then selecting **Create Transform with InstallShield** from the shortcut menu.

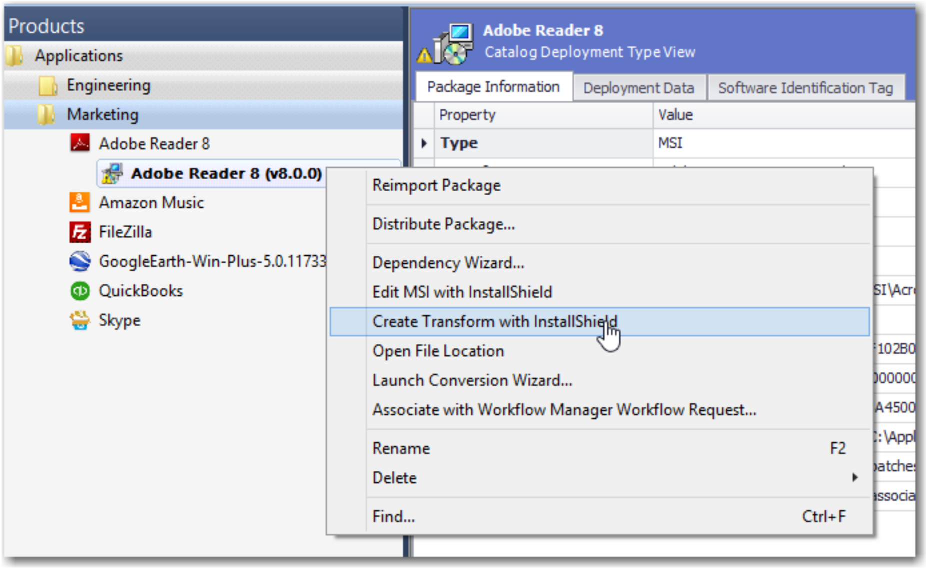
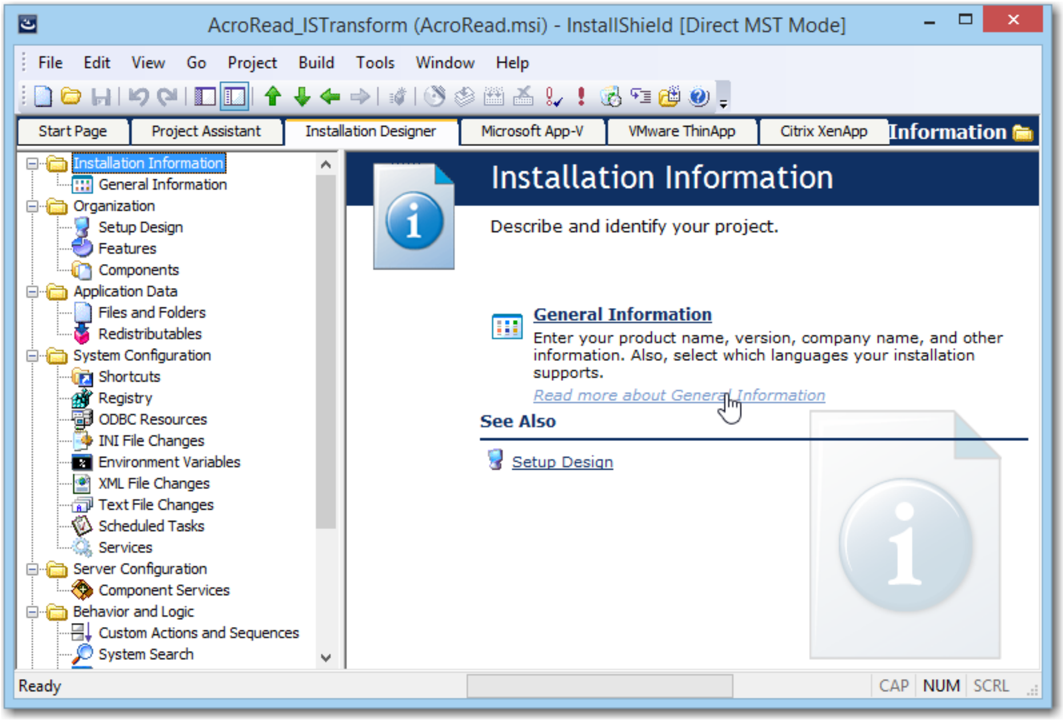


Figure 61: Create Transform with InstallShield

A new transform project (named `PackageName_ISTransform.mst`) for the selected package opens in InstallShield in Direct MST mode.



Installation Information

Describe and identify your project.

General Information

Enter your product name, version, company name, and other information. Also, select which languages your installation supports.

Read more about General Information

See Also

Setup Design

Figure 62: Transform File Open in InstallShield Direct MST Mode

After you have finished customizing this transform file in InstallShield and have saved it, you can then reimport the Windows Installer package along with its newly created transform file into the Application Catalog by right-clicking on the Windows Installer package in the Application Manager tree and selecting **Reimport Package** from the shortcut menu. This lets Application Manager know that you are done editing the transform file.

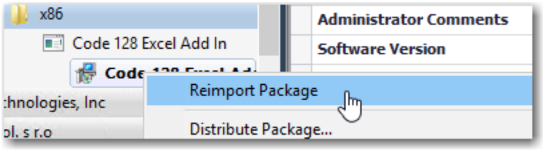


Figure 63: Reimport Package Command on Shortcut Menu

Ability to Browse to Package Location from Application Manager Tree

In AdminStudio 2016, you can quickly browse to the directory location of your source package files by right-clicking on the package in the Application Manager tree and then selecting **Open File Location** from the shortcut menu. This new option is available on both the **Catalog** and the **Test Center** tabs of Application Manager.

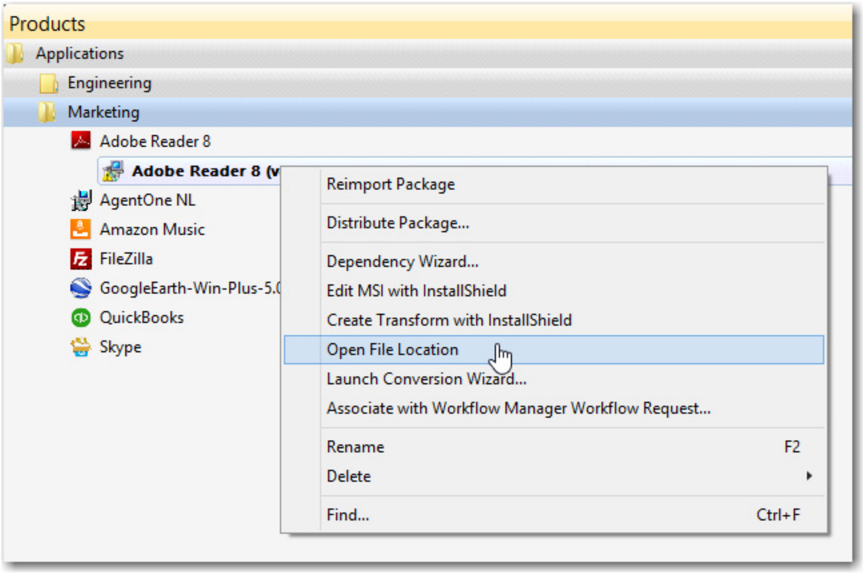
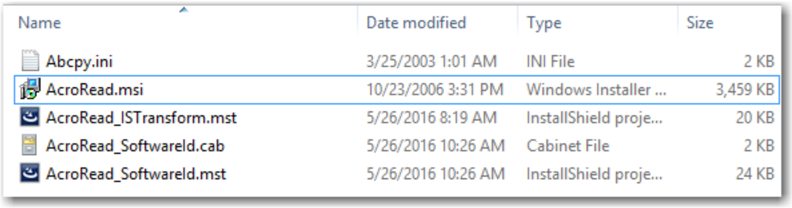


Figure 64: New “Open File Location” Command on Shortcut Menu

A new File Explorer window opens, listing the package’s source files.



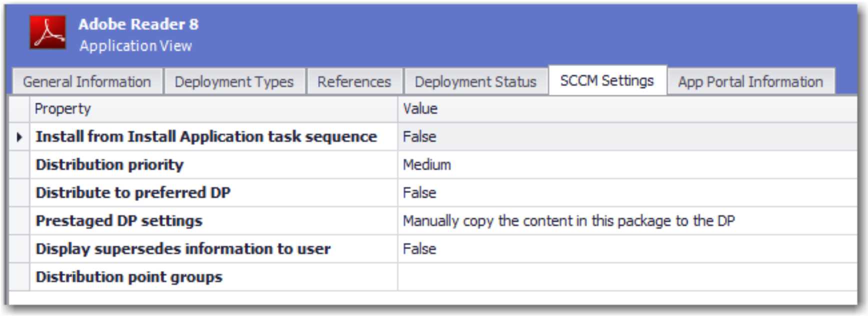
Name	Date modified	Type	Size
Abcpy.ini	3/25/2003 1:01 AM	INI File	2 KB
AcroRead.msi	10/23/2006 3:31 PM	Windows Installer ...	3,459 KB
AcroRead_ISTransform.mst	5/26/2016 8:19 AM	InstallShield proje...	20 KB
AcroRead_SoftwareId.cab	5/26/2016 10:26 AM	Cabinet File	2 KB
AcroRead_SoftwareId.mst	5/26/2016 10:26 AM	InstallShield proje...	24 KB

Figure 65: File Location of Imported Package’s Source Files

New System Center Configuration Manager Settings View

In previous releases, an application’s System Center Configuration Manager settings were listed on the **General Information** tab of the **Application View**. These settings were displayed whether or not you had configured a connection to a System Center Configuration Manager server.

In AdminStudio 2016, a new tab has been added to the **Application View** entitled **SCCM Settings** where these System Center Configuration Manager settings are now displayed. The **SCCM Settings** tab is only displayed if you have set up a connection to a System Center Configuration Manager server.

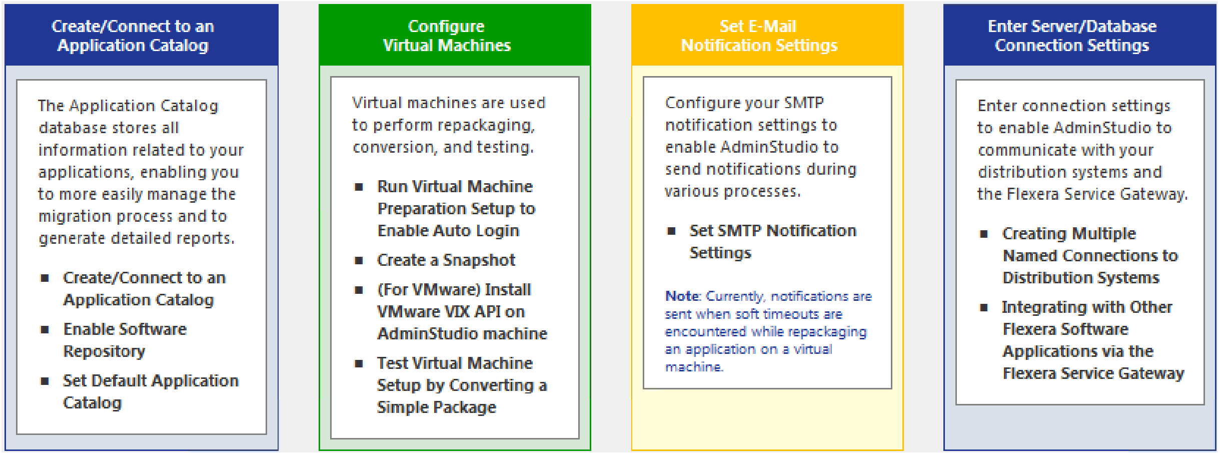


Property	Value
Install from Install Application task sequence	False
Distribution priority	Medium
Distribute to preferred DP	False
Prestaged DP settings	Manually copy the content in this package to the DP
Display supersedes information to user	False
Distribution point groups	

Figure 66: SCCM Settings Tab of Application View

Setting Up AdminStudio Infrastructure

The flowchart on the **Set Up Infrastructure** tab of the AdminStudio Start page lists the steps you need to perform before you can get started using AdminStudio.



Create/Connect to an Application Catalog

The Application Catalog database stores all information related to your applications, enabling you to more easily manage the migration process and to generate detailed reports.

- Create/Connect to an Application Catalog
- Enable Software Repository
- Set Default Application Catalog

Configure Virtual Machines

Virtual machines are used to perform repackaging, conversion, and testing.

- Run Virtual Machine Preparation Setup to Enable Auto Login
- Create a Snapshot
- (For VMware) Install VMware VIX API on AdminStudio machine
- Test Virtual Machine Setup by Converting a Simple Package

Set E-Mail Notification Settings

Configure your SMTP notification settings to enable AdminStudio to send notifications during various processes.

- Set SMTP Notification Settings

Note: Currently, notifications are sent when soft timeouts are encountered while repackaging an application on a virtual machine.

Enter Server/Database Connection Settings

Enter connection settings to enable AdminStudio to communicate with your distribution systems and the Flexera Service Gateway.

- Creating Multiple Named Connections to Distribution Systems
- Integrating with Other Flexera Software Applications via the Flexera Service Gateway

Figure 67: Set Up Infrastructure Tab of AdminStudio Start Page

These are specific one-time set-up tasks that you need to perform prior to using AdminStudio. To set up AdminStudio infrastructure, perform the following exercises:

- Creating an Application Catalog
- Configuring a Virtual Machine
- Setting E-Mail Notification Settings
- Entering Server/Database Connection Settings

Creating an Application Catalog

With AdminStudio, you manage your applications and their deployment types in an Application Catalog database, which stores all package information, including test results. This enables you to perform enterprise level data checking. You can share your Application Catalog between multiple AdminStudio users.

AdminStudio and many of its tools (such as Application Manager) require you to be connected to an Application Catalog, while others give you the option of working with packages on a local or network directory or from Microsoft System Center Configuration Manager Server.

In this exercise, you will create a new AdminStudio Application Catalog and set it as the default.

Table 3 • Create/Connect to an Application Catalog

#	Step	Instructions	Result
1.	**Create an Application Catalog**	Open AdminStudio and create a new SQL Server Application Catalog database named `AdminStudio Evaluation Catalog`, as described in Creating New Application Catalogs Using the AdminStudio Interface. ***Note** • On the **Select Software Repository Location** panel of the **Application Catalog Wizard**, do NOT select the **Enable Software Repository** option.*	AdminStudio is open and connected to a new Application Catalog named `AdminStudio Evaluation Catalog`.

Configuring a Virtual Machine

Virtual machines are used by Automated Application Converter during automated repackaging (performed during conversion to virtual applications) and when testing applications.

You need to prepare each virtual machine that you are going to use with the Automated Application Converter to perform automated repackaging or testing by doing the following:

- **Run Virtual Machine Preparation setup**—On each virtual machine you are going to use with the Automated Application Converter, run the Virtual Machine Preparation setup, an application that will enable automatic login. You need to run this application one time on all of the virtual machines that you are going to use with the Automated Application Converter.
- **Create a snapshot**—After you have run the Virtual Machine Preparation setup on a virtual machine, you need to shut it down and create a snapshot named `AutoRepack_Base`. This enables the Automated Application Converter to revert the virtual image to a clean state after each repackaging run.
- **Install VMware VIX API (VMware only)**—In order for the Automated Application Converter to perform automated repackaging, it needs to communicate with the virtualization technology that you are using. If you are using VMware virtualization technology (VMware ESX or ESXi Server or a local VMware Workstation 6.5 or later), you need to have the VMware VIX API installed on the same machine as the Automated Application Converter.

In this exercise, you will configure a virtual machine for use with Automated Application Converter.

Table 4 • Configure Virtual Machines

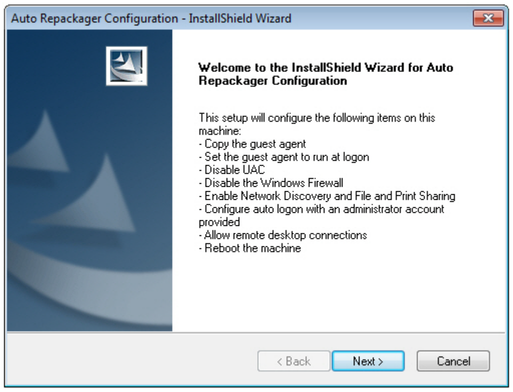
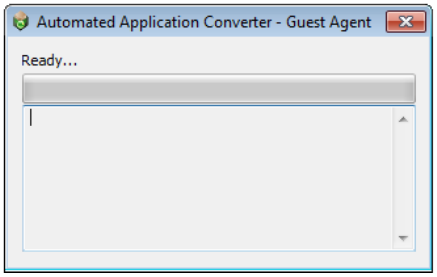
#	Step	Instructions	Result
1.	**Run Virtual Machine Preparation setup to enable auto login**	On a Microsoft Hyper-V Server image, VMware ESX/ESXi Server image, or VMware Workstation (6.5+) image, run the virtual machine preparation setup.  Auto Repackager Configuration - InstallShield Wizard **Welcome to the InstallShield Wizard for Auto Repackager Configuration** This setup will configure the following items on this machine: - Copy the guest agent - Set the guest agent to run at logon - Disable UAC - Disable the Windows Firewall - Enable Network Discovery and File and Print Sharing - Configure auto logon with an administrator account provided - Allow remote desktop connections - Reboot the machine ***Note** • For instructions, see Preparing Your Virtual Machines for Use With the Automated Application Converter.*	When you restart the virtual machine image, you are automatically logged in and `GuestAgent.exe` is launched.  Automated Application Converter - Guest Agent Ready...
2.	**Create a snapshot for repackaging**	On the prepared virtual image, create a clean snapshot named `AutoRepack_Base`. ***Note** • For instructions, see **Taking a Snapshot** in Preparing Your Virtual Machines for Use With the Automated Application Converter.*	A clean snapshot named `AutoRepack_Base` exists on the virtual machine.

Table 4 • Configure Virtual Machines

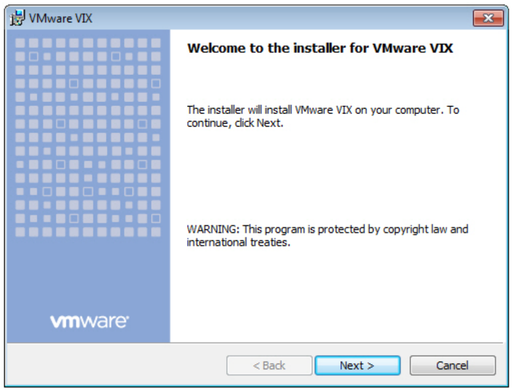
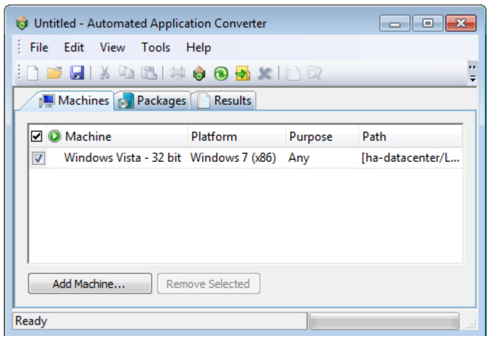
#	Step	Instructions	Result
3.	**Install VMware VIX**	If you are using a VMware ESX/ESXi Server or VMware Workstation image, you need to install the VMware VIX API on the AdminStudio machine.  **Welcome to the installer for VMware VIX** The installer will install VMware VIX on your computer. To continue, click Next. WARNING: This program is protected by copyright law and international treaties. To install the VMware VIX API on the AdminStudio machine, do one of the following: ● Install VMware Workstation on the AdminStudio machine. ● Download and install the VMware VIX API on the AdminStudio machine. You can download the VMware VIX API from the following location: http://www.vmware.com/support/developer/vix-api ***Note** • For instructions, see VMware VIX API Requirement.*	The VMware VIX API is installed successfully.

Table 4 • Configure Virtual Machines

#	Step	Instructions	Result
4.	**Add a virtual machine to Automated Application Converter**	Open Automated Application Converter by clicking **Add Virtual Machines** on the **Set Up Infrastructure** tab of the AdminStudio Start page. The **Application Conversion Project Wizard** opens. Cancel the wizard. Then open the **Machines** tab and follow the instructions in Adding Virtual Machines Using the Virtual Machine Import Wizard to add the prepared virtual machine to the **Machines** tab. ***Note •*** *When prompted, save the Automated Application Converter project file to the following directory:* `C:\Users\YOURNAME\Documents\MyProject.aacx` *All of the connection information for the virtual machine that you have set up is stored in the project file, so remember the name and location of your Automated Application Converter project file.*	A virtual machine is listed on the **Machines** tab of Automated Application Converter.  Untitled - Automated Application Converter File Edit View Tools Help Machines Packages Results ☑ Machine Platform Purpose Path ☑ Windows Vista - 32 bit Windows 7 (x86) Any [ha-datacenter/L... Add Machine... Remove Selected Ready

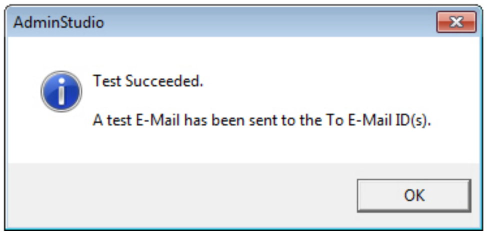
Setting E-Mail Notification Settings

To enable AdminStudio to send you e-mail notifications during various processes, you need to configure your SMTP notification settings.

Currently, e-mail notifications are sent when soft time-outs are encountered while using Automated Application Converter to repackage an application on a virtual machine.

In this exercise, you will enter the SMTP settings for e-mail notifications. This enables AdminStudio to send notifications when a soft time out is encountered during repackaging on a virtual machine by Automated Application Converter.

Table 5 • Set E-Mail Notification Settings

#	Step	Instructions	Result
1.	**Set SMTP Notification Settings**	On the **Notification Settings** tab of the AdminStudio **Options** dialog box, enter your SMTP settings for e-mail notifications. ***Note** • For instructions, see Setting E-Mail Notification Settings.*	When you click **Test** on the **Notifications Settings** tab, a successful message opens.  Test Succeeded. A test E-Mail has been sent to the To E-Mail ID(s).

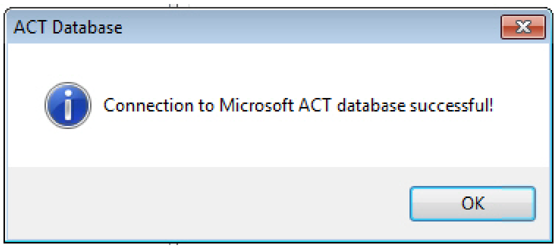
Entering Server/Database Connection Settings

In AdminStudio 2016 SP2, you can define multiple named connections to System Center Configuration Manager, Citrix XenApp, Symantec Altiris Client Management Suite, Microsoft Server App-V, JAMF Casper Suite, and AirWatch distribution systems. This enables you to both have multiple connections easily available during import and distribution, and to refer to those connection settings by name in Platform API commands.

You need to specify at least one named connection to a distribution system in order for Application Manager to import packages, distribute applications, or report on application deployment status.

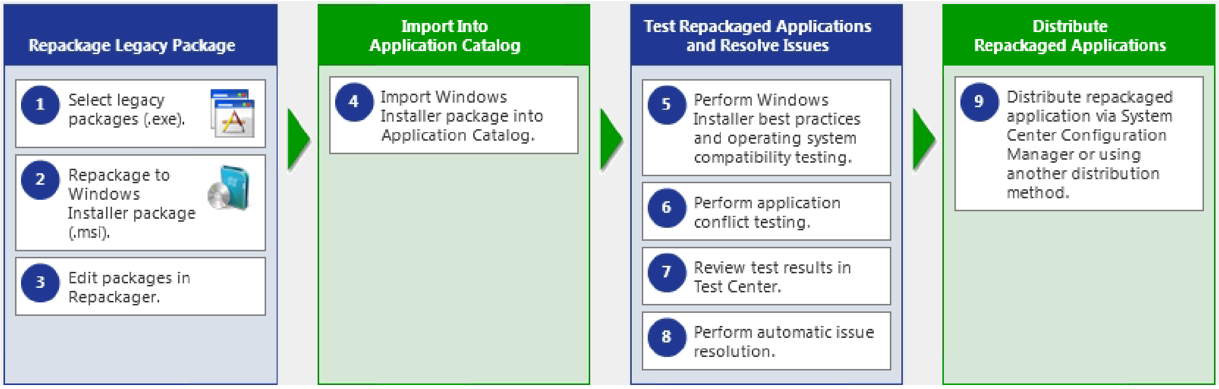
To enable AdminStudio to display data from your Microsoft ACT (Application Compatibility Toolkit) database in Test Center views and reports, you need to enter connection information for your Microsoft ACT database.

Table 6 • Enter Server Connection Settings

#	Step	Instructions	Result
1.	**Enter System Center Configuration Manager connection settings**	Open the Application Manager **Options** dialog box, and on the **Distribution System** tab, create a named connection to System Center 2012 Configuration Manager. ***Note*** *• For instructions, see Creating a New Distribution System Connection Setting*	When you click **Test** on the **Distribution System** tab, the following message is displayed: `Connection to` *`ServerName`* `Succeeded`
2.	**Entering Microsoft ACT database connection settings**	Open the Application Manager **Options** dialog box, and on the **Microsoft ACT** tab, enter Microsoft ACT database connection information. ***Note*** *• For instructions, see Entering Microsoft ACT Database Connection Settings* ***Note*** *• This is an optional step that you can perform if your organization has a Microsoft ACT database and you want to display that data in Application Manager.*	When you click **Test** on the **Microsoft ACT** tab, a successful message opens:  ACT Database Connection to Microsoft ACT database successful!

Migrating to Windows Installer

The flowchart on the **Migrate to Windows Installer** tab of the AdminStudio Start page lists the steps you need to perform to migrate a sample setup (such as an `.exe` file) to a deployable Windows Installer package.



Repackage Legacy Package

1 Select legacy packages (.exe).

2 Repackage to Windows Installer package (.msi).

3 Edit packages in Repackager.

Import Into Application Catalog

4 Import Windows Installer package into Application Catalog.

Test Repackaged Applications and Resolve Issues

5 Perform Windows Installer best practices and operating system compatibility testing.

6 Perform application conflict testing.

7 Review test results in Test Center.

8 Perform automatic issue resolution.

Distribute Repackaged Applications

9 Distribute repackaged application via System Center Configuration Manager or using another distribution method.

Figure 68: Migrate to Windows Installer Tab of AdminStudio Start Page

In this section, you will convert a sample setup named `SampleApplicationSetup.exe` to a Windows Installer Package, import it into the Application Catalog, test it for best practices, operating system compatibility, and application conflicts, and then prepare it for distribution using Distribution Wizard.

***Important** • It is preferable to repackage 32-bit applications on 32-bit operating systems. In this exercise, we will be repackaging a 32-bit application.*

To migrate a sample application to a Windows Installer package, perform the following steps:

- Repackaging a Sample Package
- Importing Packages into the Application Catalog
- Testing a Repackaged Application and Resolving Issues

Repackaging a Sample Package

In this procedure, you will repackage a sample setup, perform some minor edits in Repackager, and then build a Windows Installer package.

Table 7 • Repackage a Sample Package

#	Step	Instructions	Result

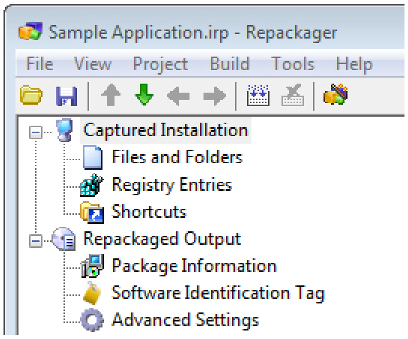
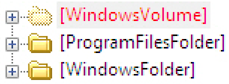
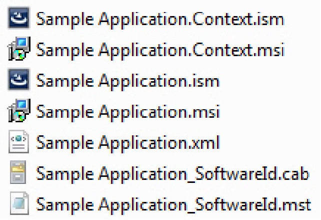
1.
2.
	Select and repackage a sample package (.exe) to a Windows Installer Package (.msi)	Use Repackaging Wizard (**Installation Monitoring** method) to repackage the following sample setup:
		`C:\AS2016EvalGuideDataFiles\ SampleApplicationSetup\ SampleApplicationSetup.exe`
		***Important** • It is recommended that you repackage this sample package on a clean machine.*
		Save the captured data in the following directory:
		`C:\Packages`
		***Note** • For instructions, see Repackaging Using the Installation Monitoring Method.*
			The captured data was converted into a Repackager project file (`SampleApplication.irp`) and opened in the Repackager interface.
			 Sample Application.irp - Repackager
			File
			View
			Project
			Build
			Tools
			Help
			Captured Installation
			Files and Folders
			Registry Entries
			Shortcuts
			Repackaged Output
			Package Information
			Software Identification Tag
			Advanced Settings

Table 7 • Repackage a Sample Package

#	Step	Instructions	Result
3.	**Edit package in Repackager and build Windows Installer package**	To exclude files that are not part of the package, open the **Files and Folders** view, right-click on the **[WindowsVolume]** folder and select **Exclude All** from the context menu. Click **Save**. ***Note** • For instructions, see Excluding All Files in a Directory.*	The **[WindowsVolume]** folder is displayed in red to indicate that it is excluded:  [WindowsVolume] [ProgramFilesFolder] [WindowsFolder]
		Open the **Repackaged Output** view and click **Build** to build a Windows Installer package. ***Note** • For instructions, see Building a Windows Installer Package.*	The Repackager project file has been converted to a Windows Installer package, located in the following directory: `C:\Packages\MSI_Package\SampleApplication.msi`
		Copy all of the files in the `C:\Packages\MSI_Package` directory (including the `SampleApplication.msi` file) to the following directory on the machine where AdminStudio is installed: `C:\AS2016EvalGuideDataFiles\SampleApplicationSetup\`	New Windows Installer package (and other associated files) are now in the same main directory as the other evaluation data files.  Sample Application.Context.ism Sample Application.Context.msi Sample Application.ism Sample Application.msi Sample Application.xml Sample Application_SoftwareId.cab Sample Application_SoftwareId.mst

Importing Packages into the Application Catalog

In this procedure, you will import Windows Installer packages into the Application Catalog, including the one you created in Repackaging a Sample Package:

Table 8 • Import Into Application Catalog

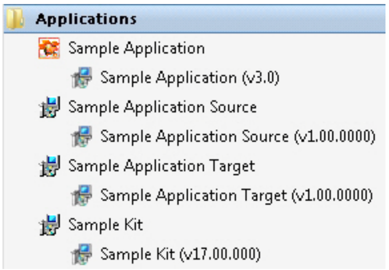
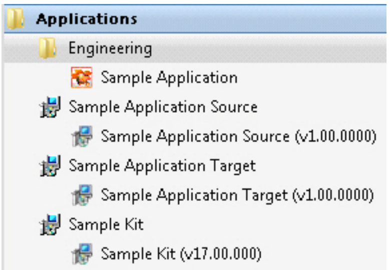
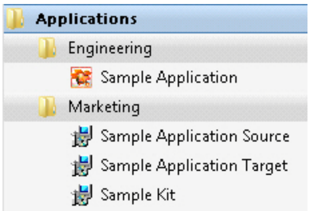
#	Step	Instructions	Result
4.	**Import Windows Installer package into Application Catalog**	Before beginning import, open the Application Manager **Options** dialog box and clear the selection of the **Automatically Execute Tests After Import** option. Next, use the Import Wizard to import all of the packages in the `AS2016EvalGuideDataFiles` directory. ● On the **Source** panel, select **Folder of multiple applications**. ● On the **Package Type Selection** panel, select the **Microsoft Windows Installer package (.msi)** option ● On the **Package Folder Selection** panel, select the `C:\AS2016EvalGuideDataFiles` directory. ● On the **Select Applications** panel, leave all of the applications selected. ● On the **Destination Group** panel, select the **Applications** folder in the tree. Do not select the **Create subgroups based on source folder structure** option. ***Note •*** *For instructions on how to import a directory of packages into the Application Catalog, see Importing a Folder of Multiple Packages.*	The packages are now listed in the Application Manager tree, each under its own Application node:  Applications Sample Application Sample Application (v3.0) Sample Application Source Sample Application Source (v1.00.0000) Sample Application Target Sample Application Target (v1.00.0000) Sample Kit Sample Kit (v17.00.000)

Table 8 • Import Into Application Catalog

<table>
<tr><th>#</th><th>Step</th><th>Instructions</th><th>Result</th></tr>
<tr><td><strong>4.</strong></td><td><strong>Import Windows Installer package into Application Catalog</strong>
(Continued)</td><td>Create a new group in the Application Manager tree named <code>Engineering</code> and move the <strong>Sample Application</strong> into that new group.
<em><strong>Note</strong> • For instructions, see Adding Groups and Organizing Applications in Application Manager.</em></td><td>The Application Manager tree should now be organized as follows:

Applications
Engineering
Sample Application
Sample Application Source
Sample Application Source (v1.00.0000)
Sample Application Target
Sample Application Target (v1.00.0000)
Sample Kit
Sample Kit (v17.00.000)</td></tr>
<tr><td></td><td></td><td>Create another new group in the Application Manager tree named <code>Marketing</code>, and move the other three applications into that group.</td><td>The Application Manager tree should now be organized as follows:

Applications
Engineering
Sample Application
Marketing
Sample Application Source
Sample Application Target
Sample Kit</td></tr>
</table>

Testing a Repackaged Application and Resolving Issues

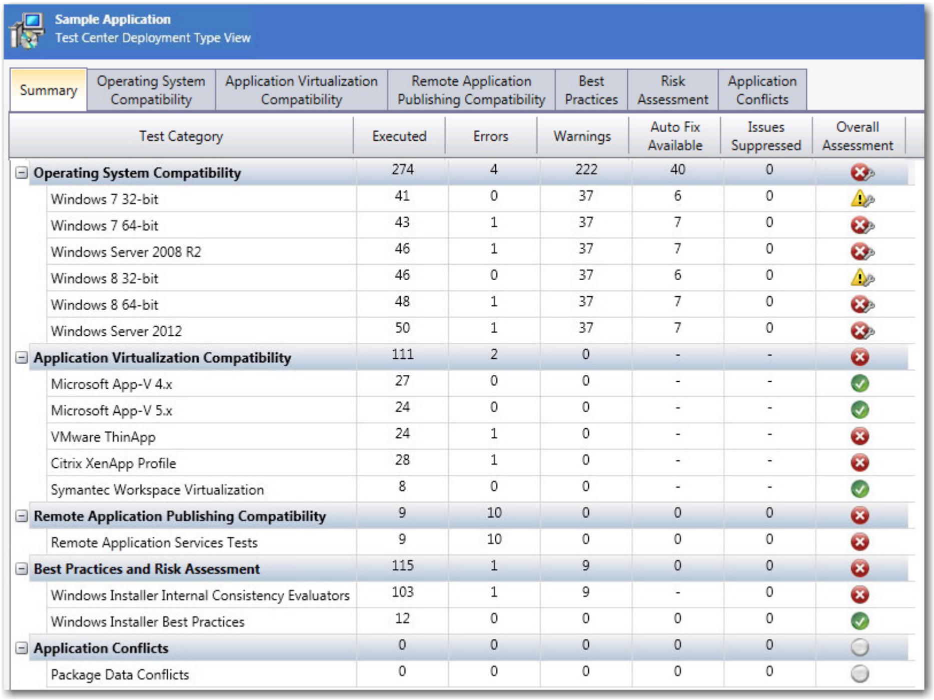
In this procedure, you will test the Windows Installer packages you imported into the Application Catalog, and then distribute a package to a network location.

Table 9 • Test and Distribute Repackaged Applications

#	Step	Instructions	Results
5.	**Perform Windows Installer best practices and OS compatibility testing**	To perform Windows Installer best practices and operating system compatibility testing on all of the packages in the Application Catalog, first select the **Test Center** tab in the Application Manager ribbon. Then, select the **Applications** group in the Application Manager tree, and click **Execute Tests**.	

Note • *For more information, see Performing Compatibility and Best Practices Testing.*

When testing is finished, view the test results by selecting a Windows Installer package in the tree to open the **Summary** tab of the **Test Center Deployment Type View**, as shown below:



Sample Application
Test Center Deployment Type View

Summary
Operating System Compatibility
Application Virtualization Compatibility
Remote Application Publishing Compatibility
Best Practices
Risk Assessment
Application Conflicts

Test Category	Executed	Errors	Warnings	Auto Fix Available	Issues Suppressed	Overall Assessment
⊟ **Operating System Compatibility**	274	4	222	40	0	✗
Windows 7 32-bit	41	0	37	6	0	⚠
Windows 7 64-bit	43	1	37	7	0	✗
Windows Server 2008 R2	46	1	37	7	0	✗
Windows 8 32-bit	46	0	37	6	0	⚠
Windows 8 64-bit	48	1	37	7	0	✗
Windows Server 2012	50	1	37	7	0	✗
⊟ **Application Virtualization Compatibility**	111	2	0	-	-	✗
Microsoft App-V 4.x	27	0	0	-	-	✓
Microsoft App-V 5.x	24	0	0	-	-	✓
VMware ThinApp	24	1	0	-	-	✗
Citrix XenApp Profile	28	1	0	-	-	✗
Symantec Workspace Virtualization	8	0	0	-	-	✓
⊟ **Remote Application Publishing Compatibility**	9	10	0	0	0	✗
Remote Application Services Tests	9	10	0	0	0	✗
⊟ **Best Practices and Risk Assessment**	115	1	9	0	0	✗
Windows Installer Internal Consistency Evaluators	103	1	9	-	0	✗
Windows Installer Best Practices	12	0	0	0	0	✓
⊟ **Application Conflicts**	0	0	0	0	0	○
Package Data Conflicts	0	0	0	0	0	○

In Test Center views, groups, applications, and packages are assigned a test status in each test group using status icons. For packages, the status icon identifies that package's test status (as described in About Status Icons). For groups and applications, Test Center considers all of the packages in that group or application, and displays the status icon for the package that has the status at the highest hierarchical level, as described in the **Hierarchical Level of Status Icons** section of the About Status Icons help topic.

Note • *For more information, see Viewing Summary Test Results.*

Table 9 • Test and Distribute Repackaged Applications

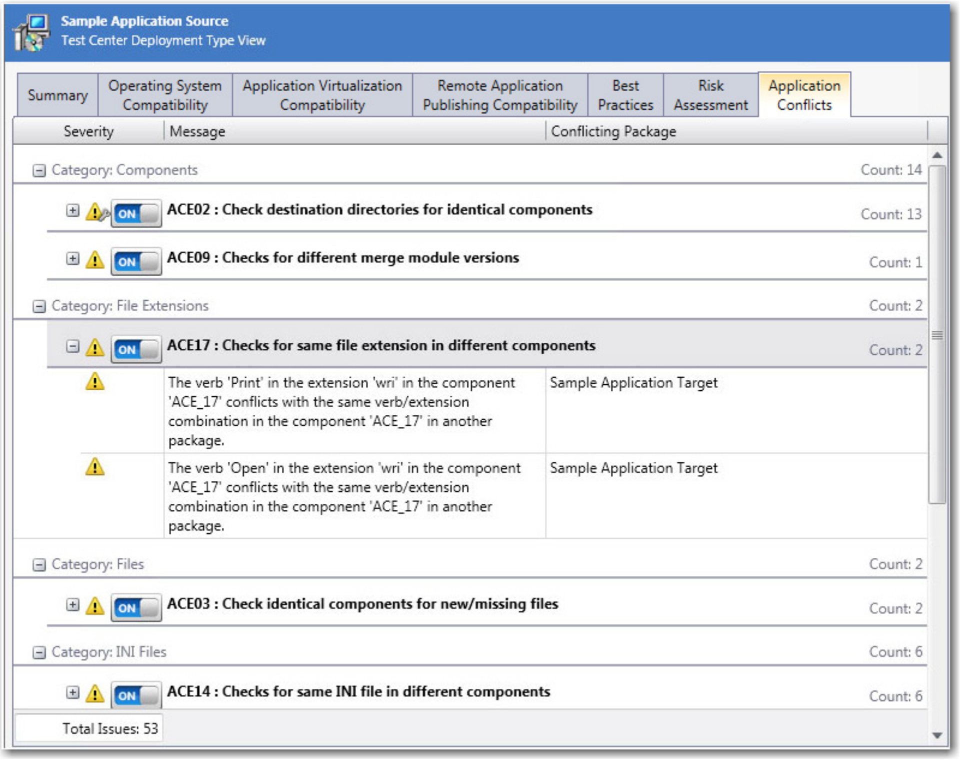
#	Step	Instructions	Results
6.	**Perform application conflict testing**	Use the Conflict Wizard to detect conflicts between the following two packages: ● **Source**: Sample Application Source ● **Target**: Sample Application Target ***Note** • For instructions, see Testing for Conflicts Between Packages.*	
7.	**Review test results in Test Center**	When application conflict testing is finished, view the test results by selecting the Sample Application Source MSI package in the tree, and then selecting the **Application Conflicts** tab of the **Test Center Deployment Type View**. Errors and warnings are listed. Click the plus sign next to a test name to view individual error/warning messages for that package:  Sample Application Source Test Center Deployment Type View Summary Operating System Compatibility Application Virtualization Compatibility Remote Application Publishing Compatibility Best Practices Risk Assessment Application Conflicts Severity Message Conflicting Package ⊟ Category: Components — Count: 14 ⊞ ⚠ ON **ACE02 : Check destination directories for identical components** — Count: 13 ⊞ ⚠ ON **ACE09 : Checks for different merge module versions** — Count: 1 ⊟ Category: File Extensions — Count: 2 ⊟ ⚠ ON **ACE17 : Checks for same file extension in different components** — Count: 2 ⚠ The verb 'Print' in the extension 'wri' in the component 'ACE_17' conflicts with the same verb/extension combination in the component 'ACE_17' in another package. — Sample Application Target ⚠ The verb 'Open' in the extension 'wri' in the component 'ACE_17' conflicts with the same verb/extension combination in the component 'ACE_17' in another package. — Sample Application Target ⊟ Category: Files — Count: 2 ⊞ ⚠ ON **ACE03 : Check identical components for new/missing files** — Count: 2 ⊟ Category: INI Files — Count: 6 ⊞ ⚠ ON **ACE14 : Checks for same INI file in different components** — Count: 6 Total Issues: 53 Next, click the suppress (ON/OFF) button next to ACE09 to suppress that test from test totals and from automatic resolution. The button toggles to the OFF position and the error icon turns to gray. ***Note** • For more information, see the following topics:* ● *Viewing Detailed Package Test Results* ● *Viewing Application Conflicts Test Results* ● *Filtering Test Results by Suppressing Errors/Warnings*	

Table 9 • Test and Distribute Repackaged Applications

#	Step	Instructions	Results
8.	**Perform automatic issue resolution**	Issues for which automatic fixes are available are identified by the Error With Fix or Warning With Fix icon: ✖ ⚠ To automatically resolve all issues for which automatic fixes are available, select the **Applications** group node in the tree and click **Resolve Issues** in the **Test Center** tab of the ribbon. ***Note** • For more information, see Performing Automatic Issue Resolution.*	Issue resolution begins, progress messages appear in the Output window, and Application Manager performs the following tasks: ● **Reruns tests**—Application Manager reruns all of the selected tests to ensure that the issues that it is going to resolve still exist in the current version of the package and its associated transforms. ● **Creates transform files**—To resolve issues, Application Manager generates fix transform files. ● **Reimports packages**—Application Manager then automatically reimports each package and its fix transform files into the Application Catalog. When issue resolution and reimporting is complete, look at the **Test Center Group View**, **Application View**, or **Deployment Type View** of the package, application, or group that you tested. You will see that the Error With Fix and Warning With Fix icons have been replaced with the status icon with the next highest level (as described in the **Hierarchical Level of Status Icons** section of the About Status Icons help topic) in that test category.

Distributing a Repackaged Application

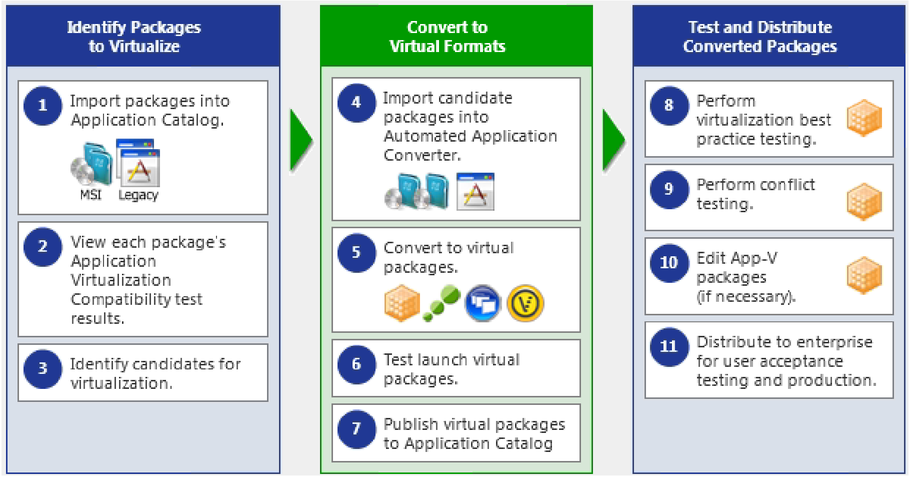
In this procedure, you will distribute a Windows Installer package to a network location.

Table 10 • Distribute a Repackaged Application

#	Step	Instructions	Results
9.	**Distribute repackaged application**	Use the Legacy Distribution Wizard to distribute the **Sample Application** Windows Installer package to a **Network** location. You open the Legacy Distribution Wizard by opening the **Catalog** tab of the ribbon, selecting the Windows Installer package node and then selecting **Distribute Package** from the context menu. For instructions on how to distribute a package to a network location, see Distributing Packages to Network Locations. ***Note •*** *To distribute an **application** to a System Center 2012 Configuration Manager, Citrix XenApp, Symantec Altiris, JAMF Casper Suite, or AirWatch, select the application node in the tree and then click the **Distribute** button in the ribbon. You must have already set up a named connection to that distribution system on the **Options** dialog box.* ***Note •*** *You can publish applications containing App-V 4.x packages and Citrix XenApp profiles to Citrix XenApp server, and can publish applications containing Windows Installer, Symantec Workspace, VMware ThinApp, or legacy installers to Symantec Altiris server.* *If an application contains a package of an unsupported deployment type, that package will be ignored.*	`SampleApplication.msi` is copied to the specified network location, making it available to your enterprise.

Migrating to Application Virtualization

The flowchart on the **Migrate to Application Virtualization** tab of the AdminStudio Start page lists the steps you need to perform to migrate your application portfolio into virtual applications that are ready for deployment within the enterprise.



Identify Packages to Virtualize

1 Import packages into Application Catalog.

2 View each package's Application Virtualization Compatibility test results.

3 Identify candidates for virtualization.

Convert to Virtual Formats

4 Import candidate packages into Automated Application Converter.

5 Convert to virtual packages.

6 Test launch virtual packages.

7 Publish virtual packages to Application Catalog

Test and Distribute Converted Packages

8 Perform virtualization best practice testing.

9 Perform conflict testing.

10 Edit App-V packages (if necessary).

11 Distribute to enterprise for user acceptance testing and production.

Figure 69: Migrate to Application Virtualization Tab of AdminStudio Start Page

In this section, you will use the Automated Application Converter to convert Windows Installer packages to virtual packages, and then test and distribute the virtual packages.

To migrate your application portfolio into virtual applications, perform the following steps:

- Identifying Packages to Virtualize
- Converting to Virtual Formats
- Testing and Distributing Converted Packages

Identifying Packages to Virtualize

In this procedure, you will import packages into the Application Catalog and identify the packages you want to virtualize.

Table 11 • Identify Packages to Virtualize

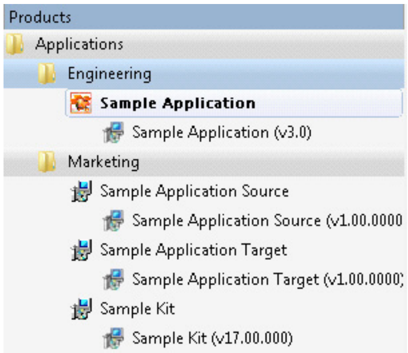
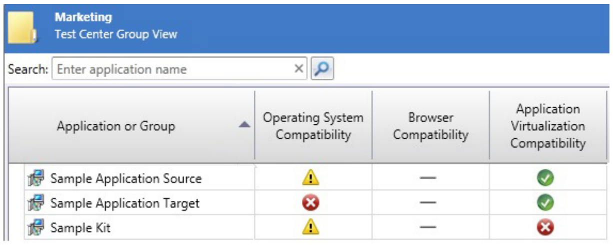
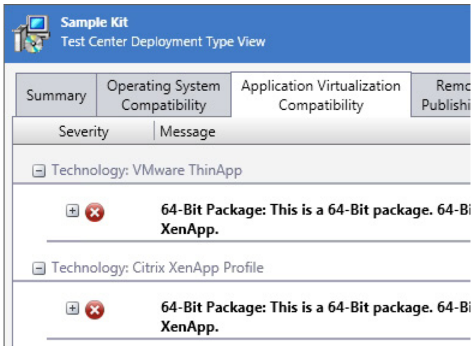
#	Step	Instructions	Results
1.	**Import packages into Application Catalog.**	Open Application Manager and locate the packages that you imported in Importing Packages into the Application Catalog.	The four Windows Installer packages are listed in the Application Manager tree:  Products Applications Engineering **Sample Application** Sample Application (v3.0) Marketing Sample Application Source Sample Application Source (v1.00.0000 Sample Application Target Sample Application Target (v1.00.0000) Sample Kit Sample Kit (v17.00.000)
2.	**View package's Application Virtualization Compatibility test results.**	To view application virtualization compatibility test results, open the **Test Center** tab, and then select the **Marketing** group in the tree to open the **Test Center Group View**.	The packages' application virtualization compatibility test results are listed in the **Application Virtualization Compatibility** column of the **Test Center Group View**.  Marketing Test Center Group View Search: Enter application name Application or Group | Operating System Compatibility | Browser Compatibility | Application Virtualization Compatibility Sample Application Source | ⚠ | — | ✓ Sample Application Target | ✗ | — | ✓ Sample Kit | ⚠ | — | ✗

Table 11 • Identify Packages to Virtualize

#	Step	Instructions	Results
3.	**Identify candidates for virtualization.**	You will notice that in the **Application Virtualization Compatibility** column of the **Test Center Group View**, two of the packages have a Ready status (Sample Application Source and Sample Application Target), while the other has an Error status (Sample Kit). Therefore, for the package with the Error status, open the **Application Virtualization Compatibility** tab of the **Test Center Deployment Type** view to view the error messages.	You will notice that the errors were generated because the packages are 64-bit packages, and 64-bit packages are not supported by ThinApp or XenApp.  Sample Kit Test Center Deployment Type View Summary Operating System Compatibility Application Virtualization Compatibility Remo Publishi Severity | Message Technology: VMware ThinApp 64-Bit Package: This is a 64-Bit package. 64-Bi XenApp. Technology: Citrix XenApp Profile 64-Bit Package: This is a 64-Bit package. 64-Bi XenApp. However, since 64-bit packages are supported by Microsoft App-V, this package is ready for conversion to App-V format.

Converting to Virtual Formats

In this procedure, you will use Automated Application Converter to convert Windows Installer packages to App-V packages.

Table 12 • Convert to Virtual Formats

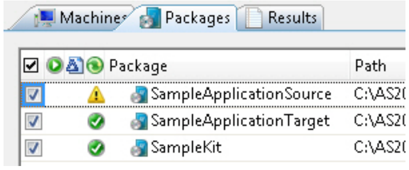
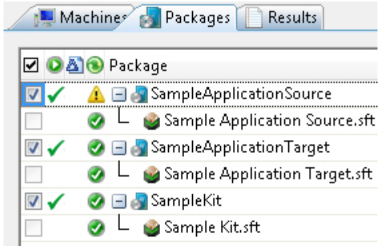
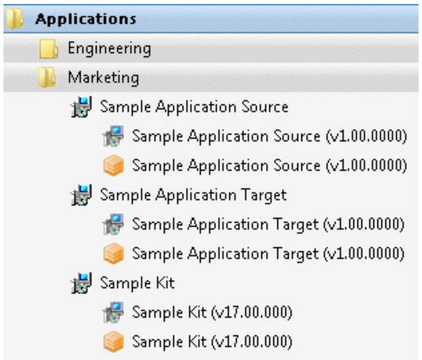
#	Step	Instructions	Results
4.	**Import candidate packages into Automated Application Converter.**	Open Automated Application Converter, open the existing project you created in Configuring a Virtual Machine, and use the **Package Import Wizard** to add the packages that you identified in Identifying Packages to Virtualize. ***Note •*** *For instructions, see Selecting Packages from an AdminStudio Application Catalog.*	Packages are listed on the **Packages** tab.  ☑ Package | Path ☑ ⚠ SampleApplicationSource | C:\AS20 ☑ ✓ SampleApplicationTarget | C:\AS20 ☑ ✓ SampleKit | C:\AS20

Table 12 • Convert to Virtual Formats

#	Step	Instructions	Results
5.	**Convert to virtual packages.**	First. open the **Project Options** dialog box and make sure that the **Package Creation** property is set to **App-V 4.6 with AdminStudio**. Then, use the Application Conversion Wizard to convert the packages to App-V 4.x applications. ***Note •*** *For instructions, see Performing a Conversion Using the Application Conversion Wizard.* ***Important •*** *Make sure that you have already performed the steps in Configuring a Virtual Machine before beginning this step.*	When conversion is complete, each virtual package is listed in a tree structure under its original package on the **Packages** tab.  Machines | Packages | Results ☑ Package ☑ ✓ SampleApplicationSource □ Sample Application Source.sft ☑ ✓ SampleApplicationTarget □ Sample Application Target.sft ☑ ✓ SampleKit □ Sample Kit.sft
6.	**Test launch virtual packages.**	Test the virtual packages by launching them on a virtual machine. ***Note •*** *For instructions, see Performing Automated Testing of App-V Packages.*	Virtual packages launch successfully.
7.	**Publish virtual packages to Application Catalog.**	Import the new virtual packages into the Application Catalog. ***Note •*** *For instructions, see Importing a Single Package File.*	The virtual packages are listed in the Application Manager tree under their associated Application.  Applications Engineering Marketing Sample Application Source Sample Application Source (v1.00.0000) Sample Application Source (v1.00.0000) Sample Application Target Sample Application Target (v1.00.0000) Sample Application Target (v1.00.0000) Sample Kit Sample Kit (v17.00.000) Sample Kit (v17.00.000) ***Note •*** *You may need to click Ctrl + R to refresh the Application Catalog tree.*

Testing and Distributing Converted Packages

In this procedure, you will validate the converted packages, perform conflict analysis against other packages, resolve any issues found, and distribute the packages.

Table 13 • Test and Distribute Converted Packages

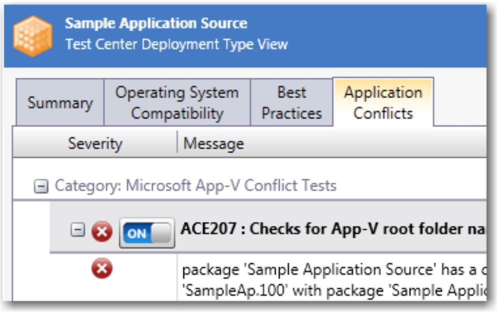
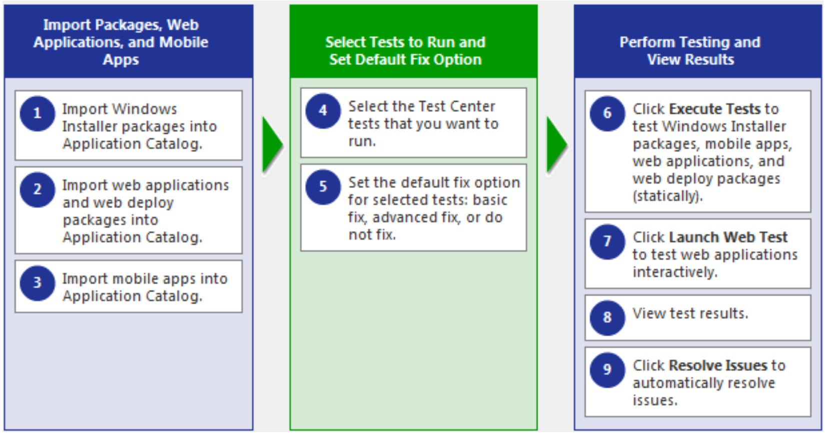
#	Step	Instructions	Results
8.	**Perform virtualization best practices testing.**	By default, App-V best practices testing is performed during import (as described in Import Options). To perform this testing manually, first select the **Test Center** tab in the Application Manager ribbon. Then, select one of the App-V packages in the Application Manager tree, and click **Execute Tests**. Messages appear in the Output Window. ***Note •*** *For more information, see Performing Compatibility, Best Practices, and Risk Assessment Testing.*	When testing is complete, results are displayed on the **Summary** tab of the **Test Center Deployment Type View**.
9.	**Perform conflict testing.**	In Application Manager, perform conflict testing of the **Sample Application Source** App-V package against the **Sample Application Target** App-V package. ***Note •*** *For instructions, see Testing for Conflicts Between Packages.*	Conflict analysis results are listed in the **Output** window and in the **Conflicts** view. An ACE207 error is detected: `Package ‘Sample Application Source’ has a conflicting root Directory ‘SampleAp.100’ with package ‘Sample Application Target’.` Also, the error is listed on the **Application Conflicts** tab of the **Test Center Deployment Type View** for that package:  Sample Application Source Test Center Deployment Type View Summary Operating System Compatibility Best Practices Application Conflicts Severity Message Category: Microsoft App-V Conflict Tests ON ACE207 : Checks for App-V root folder na package 'Sample Application Source' has a c 'SampleAp.100' with package 'Sample Applic

Table 13 • Test and Distribute Converted Packages

#	Step	Instructions	Results
10.	**Edit App-V packages (if necessary).**	To resolve the error that was found during testing on **Sample Application Source** App-V package, select it on the **Catalog** tab of the Application Manager tree and select **Edit with Virtual Package Editor**. In the Virtual Package Editor’s **General Information** view, change the **Root Folder Name** property to **SampleAp.200** and click **Save**. ***Note*** • *For instructions, see Using the Virtual Package Editor.* Return to Application Manager and reimport the edited package.	The edited App-V package is imported into the Application Catalog. ***Note*** • *When you rerun the conflict testing that you ran in the previous step, no error will be generated.*
11.	**Distribute to enterprise for user acceptance testing and production.**	Distribute this tested App-V package to a **Network** location. ***Note*** • *For instructions on how to distribute a package, see Distributing Packages to Network Locations.* ***Tip*** • *Distribution Wizard also supports deploying applications to System Center Configuration Manager (Current Branch), System Center 2012 Configuration Manager, System Center 2007 Configuration Manager, Citrix XenApp Server, JAMF Casper Suite, AirWatch Server, Microsoft App-V Server, and Symantec Altiris distribution systems. For more information, see Distributing Applications.*	The selected App-V package is copied to the specified network location, making it available to your enterprise.

Testing for Application Compatibility

The flowchart on the **Test for Application Compatibility** tab of the AdminStudio Start page outlines how to use Application Manager to test packages for compatibility with Windows 7, Windows 8, Windows 10, Windows Server 2008 R2, Windows Server 2012, and Windows Server 2016 platforms, as well as to test web applications for compatibility with Internet Explorer (9, 10, 11) and Microsoft Edge.



Import Packages, Web Applications, and Mobile Apps

1 Import Windows Installer packages into Application Catalog.

2 Import web applications and web deploy packages into Application Catalog.

3 Import mobile apps into Application Catalog.

Select Tests to Run and Set Default Fix Option

4 Select the Test Center tests that you want to run.

5 Set the default fix option for selected tests: basic fix, advanced fix, or do not fix.

Perform Testing and View Results

6 Click **Execute Tests** to test Windows Installer packages, mobile apps, web applications, and web deploy packages (statically).

7 Click **Launch Web Test** to test web applications interactively.

8 View test results.

9 Click **Resolve Issues** to automatically resolve issues.

Figure 70: Test for Application Compatibility Tab of AdminStudio Start Page

In this section, you will test some packages for operating system compatibility fix issues that were found. You will also test web applications, both statically and dynamically.

To perform application compatibility testing, perform the following steps:

- Importing Packages, Web Applications, and Mobile Apps
- Selecting Tests to Run and Setting Default Fix Option
- Performing Testing and Viewing Results

Importing Packages, Web Applications, and Mobile Apps

In this procedure, you will load the packages to test and select the reports to run.

Table 14 • Importing Packages and Web Applications

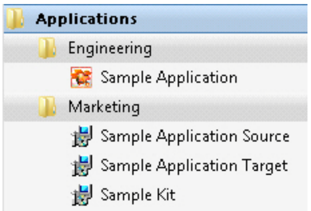
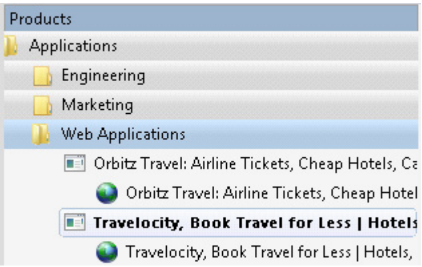
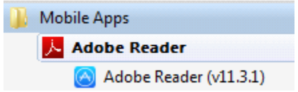
#	Step	Instructions	Result
1.	**Import Windows Installer packages into Application Catalog.**	For this exercise, we will test Windows Installer packages that were imported into the Application Catalog earlier in this guide in Importing Packages into the Application Catalog. ***Note •** For instructions on how to import a directory of packages into the Application Catalog, see Importing a Folder of Multiple Packages.*	The Application Manager tree should now be organized as follows:  Applications Engineering Sample Application Marketing Sample Application Source Sample Application Target Sample Kit
2.	**Import web applications into Application Catalog.**	First, open the **Import Options > General** tab of the Application Manager **Options** dialog box, and clear the selection of the **Automatically Execute Tests After Import** option. Next, create a new group in the Application Manager tree named **Web Applications**. Then import the following web applications into the **Web Applications** folder, as described in Importing a Deployed Web Applications. ● **Orbitz** at: http://www.orbitz.com ● **Travelocity** at: http://www.travelocity.com Because these web applications do not require a login to access, leave the **User name** and **Password** fields on the **Web Site Details** panel blank.	The web applications are listed in the Application Manager tree:  Products Applications Engineering Marketing Web Applications Orbitz Travel: Airline Tickets, Cheap Hotels, Ca Orbitz Travel: Airline Tickets, Cheap Hotel **Travelocity, Book Travel for Less | Hotels** Travelocity, Book Travel for Less | Hotels,

Table 14 • Importing Packages and Web Applications

#	Step	Instructions	Result
3.	**Import a mobile app into the Application Catalog**	Create a new group in the Application Manager tree named **Mobile Apps**. Then, import an Apple iOS mobile app from the Apple App Store, as described in Importing Public Store Mobile Apps.	The iOS public store mobile app is listed in the Application Manager tree.  Mobile Apps **Adobe Reader** Adobe Reader (v11.3.1)

Selecting Tests to Run and Setting Default Fix Option

In this procedure, you will select the Operating System Compatibility and Browser Compatibility tests to run and set default fix options.

Table 15 • Selecting Tests to Run and Setting Default Fix Option

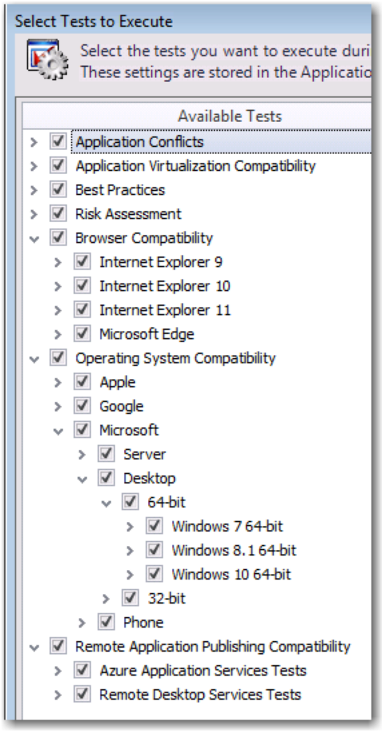
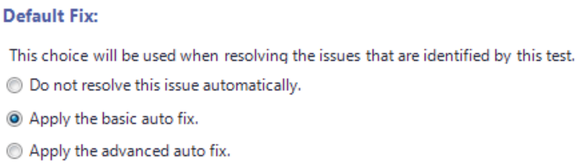
#	Step	Instructions	Result
4.	**Select the operating system and browser compatibility tests that you want to run.**	Select the **Operating System Compatibility** and **Browser Compatibility** tests that you want to run, as described in Selecting Tests to Execute. ***Tip •*** *To speed up testing, it is recommended that you select just the operating systems and browser versions that are being used in your organization.*	The **Operating System Compatibility** and **Browser Compatibility** tests that you want to run are selected on the **Select Tests to Execute** dialog box.  Select Tests to Execute Select the tests you want to execute duri These settings are stored in the Applicatio Available Tests ☑ Application Conflicts ☑ Application Virtualization Compatibility ☑ Best Practices ☑ Risk Assessment ☑ Browser Compatibility ☑ Internet Explorer 9 ☑ Internet Explorer 10 ☑ Internet Explorer 11 ☑ Microsoft Edge ☑ Operating System Compatibility ☑ Apple ☑ Google ☑ Microsoft ☑ Server ☑ Desktop ☑ 64-bit ☑ Windows 7 64-bit ☑ Windows 8.1 64-bit ☑ Windows 10 64-bit ☑ 32-bit ☑ Phone ☑ Remote Application Publishing Compatibility ☑ Azure Application Services Tests ☑ Remote Desktop Services Tests

Table 15 • Selecting Tests to Run and Setting Default Fix Option

#	Step	Instructions	Result
5.	**Set the default fix option for selected tests: basic fix, advanced fix, or do not fix.**	Some of the tests in the **Operating System Compatibility** and **Browser Compatibility** test group let you specify whether to perform a basic or advanced fix when you automatically resolve issues, as described in Setting Automatic Fix Preferences for Operating System Compatibility and Browser Compatibility Tests. On the **Select Tests to Execute** dialog box, review the **Default Fix** section of several of your selected tests. For this exercise, it is not necessary to make any changes to the **Default Fix** settings.	A **Default Fix** selection is made for all **Operating System Compatibility** and **Browser Compatibility** tests.  **Default Fix:** This choice will be used when resolving the issues that are identified by this test. ○ Do not resolve this issue automatically. ● Apply the basic auto fix. ○ Apply the advanced auto fix.

Performing Testing and Viewing Results

In this procedure, you will test packages and web applications for operating system and browser compatibility, view test results, and automatically fix issues.

Table 16 • Performing Testing and Viewing Results

#	Step	Instructions	Result

6.	**Click Execute Tests to test Windows Installer packages, mobile apps, and web applications (statically).**	First select the **Test Center** tab in the Application Manager ribbon
		Then, select the **Applications** group in the Application Manager tree, and click **Execute Tests**.
		Note • *For more information, see Performing Compatibility, Best Practices, and Risk Assessment Testing.*
			Messages are listed in the Output Window. When testing is complete, the following message is displayed:
			`Testing finished at: Thursday, June 23, 2016 - 13:47:04`
			`Tested 6 packages of 6.`

Table 16 • Performing Testing and Viewing Results

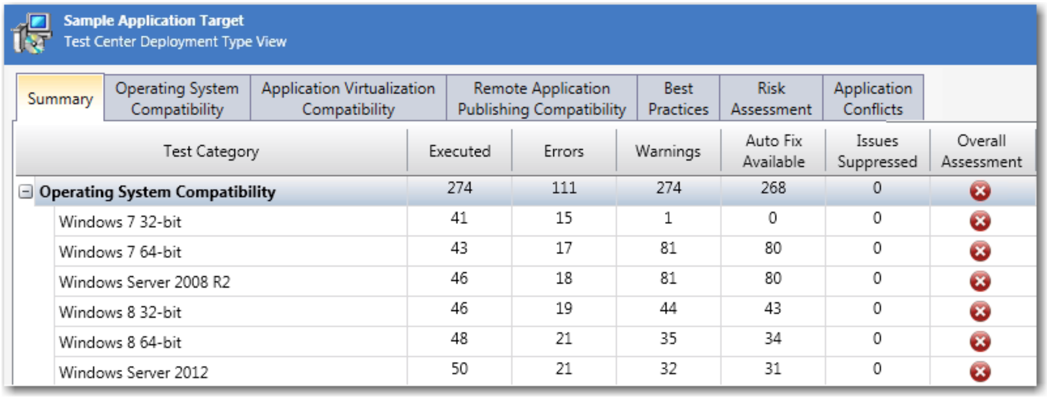
#	Step	Instructions	Result

7.	**Click Launch Web Test to test web applications interactively.**	To interactively, dynamically test a web application, select a web application node in the tree and click **Launch Web Test**.
		***Note** • For more information, see Performing Dynamic Testing of Web Applications.*
		Application Manager launches the web application in your browser. Then, as you perform tasks and navigate around the web application, Application Manager records any warnings or errors that are encountered while using that version of the browser.
		When you have finished testing, close the browser window.
		***Tip** • You should always use dynamic testing when a web application requires a login to access.*
		***Tip** • As each page loads, Application Manager begins testing. Links on each page do not become active until testing is complete on that page, so you may have to wait several seconds before proceeding.*
			Messages are listed in the Output Window. When testing is complete, the following message is displayed:
			`Testing finished at: Thursday, June 23, 2016 - 13:58:08`
			`Completed testing package(s).`

Table 16 • Performing Testing and Viewing Results

#	Step	Instructions	Result
8.	**View test results.**	When testing is finished, view the test results by selecting a Windows Installer package or web application in the tree to open the **Summary** tab of the **Test Center Deployment Type View**.	

Test result totals are listed in the **Operating System Compatibility** test group:

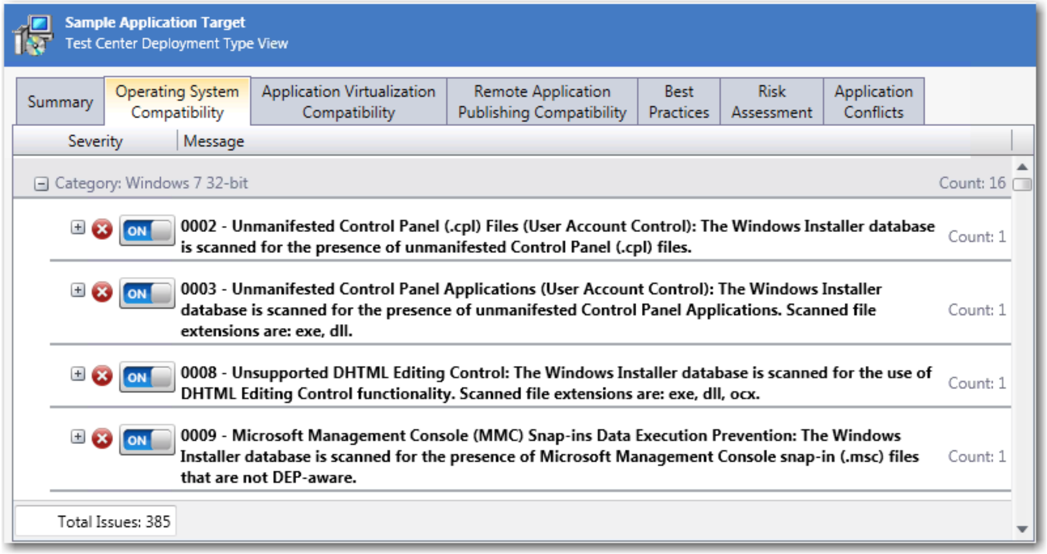


Sample Application Target
Test Center Deployment Type View

Summary
Operating System Compatibility
Application Virtualization Compatibility
Remote Application Publishing Compatibility
Best Practices
Risk Assessment
Application Conflicts

Test Category	Executed	Errors	Warnings	Auto Fix Available	Issues Suppressed	Overall Assessment
⊟ **Operating System Compatibility**	274	111	274	268	0	⊗
Windows 7 32-bit	41	15	1	0	0	⊗
Windows 7 64-bit	43	17	81	80	0	⊗
Windows Server 2008 R2	46	18	81	80	0	⊗
Windows 8 32-bit	46	19	44	43	0	⊗
Windows 8 64-bit	48	21	35	34	0	⊗
Windows Server 2012	50	21	32	31	0	⊗

Detailed test results are listed on the **Operating System Compatibility** tab:



Sample Application Target
Test Center Deployment Type View

Summary
Operating System Compatibility
Application Virtualization Compatibility
Remote Application Publishing Compatibility
Best Practices
Risk Assessment
Application Conflicts
Severity
Message

⊟ Category: Windows 7 32-bit — Count: 16

⊞ ⊗ ON — **0002 - Unmanifested Control Panel (.cpl) Files (User Account Control): The Windows Installer database is scanned for the presence of unmanifested Control Panel (.cpl) files.** — Count: 1

⊞ ⊗ ON — **0003 - Unmanifested Control Panel Applications (User Account Control): The Windows Installer database is scanned for the presence of unmanifested Control Panel Applications. Scanned file extensions are: exe, dll.** — Count: 1

⊞ ⊗ ON — **0008 - Unsupported DHTML Editing Control: The Windows Installer database is scanned for the use of DHTML Editing Control functionality. Scanned file extensions are: exe, dll, ocx.** — Count: 1

⊞ ⊗ ON — **0009 - Microsoft Management Console (MMC) Snap-ins Data Execution Prevention: The Windows Installer database is scanned for the presence of Microsoft Management Console snap-in (.msc) files that are not DEP-aware.** — Count: 1

Total Issues: 385

Click the Suppress (ON/OFF) button to suppress any issues that are not important to your organization.

***Note •** For more information, see Viewing Operating System Compatibility Test Results and Viewing Browser Compatibility Test Results.*

Table 16 • Performing Testing and Viewing Results

#	Step	Instructions	Result
9.	**Click Resolve Issues to automatically resolve issues.**	Issues for which automatic fixes are available are identified by the Error With Fix or Warning With Fix icon: ✖ ⚠ To automatically resolve all issues for which automatic fixes are available, select the **Applications** group node in the tree and click **Resolve Issues** in the **Test Center** tab of the ribbon. ***Note •** For more information, see Performing Automatic Issue Resolution*.	Issue resolution begins, progress messages appear in the Output window, and Application Manager performs the following tasks: ● **Reruns tests**—Application Manager reruns all of the selected tests to ensure that the issues that it is going to resolve still exist in the current version of the package and its associated transforms. ● **Creates transform files**—To resolve issues, Application Manager generates fix transform files. ● **Reimports packages**—Application Manager then automatically reimports each package and its fix transform files into the Application Catalog. When issue resolution and reimporting is complete, look at the **Test Center Group View**, **Application View**, or **Deployment Type View** of the package, application, or group that you tested. You will see that the Error With Fix and Warning With Fix icons have been replaced with the status icon with the next highest level (as described in the **Hierarchical Level of Status Icons** section of the About Status Icons help topic) in that test category.