Legal Information

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

Copyright Notice

Copyright © 2017 Flexera Software LLC. All Rights Reserved.

This publication contains proprietary and confidential information and creative works owned by Flexera Software LLC and its licensors, if any. Any use, copying, publication, distribution, display, modification, or transmission of such publication in whole or in part in any form or by any means without the prior express written permission of Flexera Software LLC is strictly prohibited. Except where expressly provided by Flexera Software LLC in writing, possession of this publication shall not be construed to confer any license or rights under any Flexera Software LLC intellectual property rights, whether by estoppel, implication, or otherwise.

All copies of the technology and related information, if allowed by Flexera Software LLC, must display this notice of copyright and ownership in full.

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 ................................................. 7

Getting Started ................................................................. 8
  About Evaluating AdminStudio .............................................. 8
  Purpose of Evaluation Guide ................................................. 10
  Organization of Evaluation Guide ......................................... 10
  Evaluation Guide Data Files ................................................ 13

New Features in AdminStudio 2016 SP2 ..................................... 14
  Support for PowerShell Wrapped Packages .............................. 14
  Setting PowerShell Wrap Options ......................................... 15
  PowerShell Script Template File .......................................... 17
  Wrapping a Windows Installer (.msi) or Installer Executable (.exe) 20
  Importing PowerShell Wrapped Packages ............................... 24
  Viewing a PowerShell Wrapped Package’s Bundled Child Applications 24
  Testing PowerShell Wrapped Packages .................................. 25
  PowerShell Wrapped Packages Report .................................. 26
  Distributing PowerShell Wrapped Packages ............................ 26
  Virtualizing PowerShell Wrapped Packages ............................. 27
  Testing PowerShell Wrapped Packages on a Virtual Machine ...... 27
  Customizing the PowerShell Installer .................................... 27
  Editing a PowerShell-Wrapped Package ................................ 28
  API Support for Performing PowerShell Wrapping: Invoke-ASPowerShellWrap 30

Support for Additional Types of Complex Installer Executables .......... 30
  Using Invoke-ASUpdatePackage to Extract Details on Bundled Child Packages of Imported Packages 32

Java Detection Tests and Java Dependency Reports for Desktop Applications 33
Best Practices and Risk Assessment Tests Split into Two Categories 35
New Windows Server 2016 Compatibility Tests in Test Center ........ 36
Support for System Center Configuration Manager (Current Branch) 36
Ability to Add Destination Group During System Center Configuration Manager Distribution 37
Ability to Select Connection on Microsoft Configuration Manager Deployments Report 38
Contents

New Features in AdminStudio 2016 SP1. ................................................................. 40
InstaShield 2016 SP2 ................................................................. 39

New Features in AdminStudio 2016 ................................................................. 49

- Support for the Microsoft UWP App Package (.appx) Format ........................................... 40
  - Support for Importing UWP App Packages ................................................................. 41
  - Support for Testing UWP App Packages ...................................................................... 41
  - Support for Distributing UWP App Packages to System Center Configuration Manager .... 41
- New API Commands to Set and Get Package Distribution Properties .................................. 42
- Support for the Microsoft UWP App Package (.appx) Format ........................................... 40
- InstallShield 2016 SP2 ................................................................................................. 39
  - Enhanced Automation Capabilities to Streamline Application Import and Testing .......... 72
  - Ability to Set Temporary Directory Location .............................................................. 39
  - New Mobile Risk Assessment Tests ............................................................................. 42
  - New OS Compatibility Tests ....................................................................................... 42
  - Report Center Enhancements ...................................................................................... 43
  - New Devices in Device Compatibility Reports ............................................................ 43
  - New Mac Desktop OS Application Compatibility Dashboard Report ......................... 44
  - Mac Installer Best Practices Report ............................................................................ 46
  - Applications Are Not Automatically Tested Upon Import ........................................... 47
  - Default ACE and ICE Tests are No Longer Specified on the ACE Tests Tab of Options Dialog Box ................................................................. 48
- New Features in AdminStudio 2016 .................................................................................. 50
  - Import and Test Mac OS X Desktop Applications ........................................................ 49
  - Inspect Bundled Applications in DMG and PKG Installers ........................................... 51
  - Customize Apple Installer Package PKG Installer Settings ........................................ 52
  - Import Links to OS X Desktop Applications in the Mac App Store ................................ 54
  - New Field That Identifies Genre of Mac Public Store Apps (Both iOS and OS X) ........... 55
  - Test Mac OS X Desktop Applications ........................................................................ 55
  - Combined Testing of Child Applications of DMG and PKG Files ................................. 56
  - Publish Mac OS X Desktop Applications to Casper With Integration to App Portal ........ 57
  - Setting Up a Connection to a Casper Server ................................................................ 58
  - Automatically Creating App Portal Catalog Items for DMG and PKG Applications .......... 59
  - Viewing Casper Deployment Data for an Application ................................................... 60
  - Product Name Displayed in Casper Listed on the Catalog Deployment Type View .......... 63
- Ability to Inspect and Test Complex Installer Executables .................................................. 64
  - Bundled Windows Installer Packages Now Listed on New Bundled Packages Tab ........... 64
  - Can Perform Testing of Child Windows Installer Packages of Complex Installer Executables ................................................................................................................................. 65
- Improved FlexNet Manager Suite Integration ..................................................................... 66
  - Ability to Search Application Catalog for Unrecognized Applications .......................... 66
  - Ability to Create Local Flexera Identifier Entries for Internal or Repackaged Applications ................................................................................................................................. 69
  - Ability to Enter FlexNet Manager Suite URL ................................................................ 72
- Enhanced Automation Capabilities to Streamline Application Import and Testing ............... 72
  - Applications Are Automatically Tested Upon Import ................................................... 73
  - Enhancements to Package Auto Import Feature ......................................................... 74
  - Can Monitor Multiple Directories Simultaneously ....................................................... 74
  - Can Select File Types to Monitor ................................................................................. 77
  - Can Perform Batch Auto Import of Microsoft Security Patch Files and iOS Enterprise Policy Files ................................................................. 77
  - Enhanced Control When Using Import Wizard to Perform Batch Import from a Directory ................................................................. 78

Can Perform Testing of Child Windows Installer Packages of Complex Installer Executables
Automated Application Converter Enhancements .......................... 79
Automated Application Converter for Repackaging Now Included in Professional and Enterprise Editions .......................... 79
Ability to Edit Default Automated Application Converter Settings From Application Manager .......................... 80
Can Edit Conversion Settings From the Conversion Wizard on a Per-Run Basis .......................... 82
New Wizard to Install and Launch Windows Installer Packages on a Virtual Machine for Testing .......................... 84
Productivity Shortcuts ................................................... 88
New Shortcut to Automatically Create Transform for Selected Package in InstallShield .......................... 88
Ability to Browse to Package Location from Application Manager Tree ........................................ 90
New System Center Configuration Manager Settings View ........................................ 91

Setting Up AdminStudio Infrastructure ........................................ 92
Creating an Application Catalog ........................................ 93
Configuring a Virtual Machine ........................................ 93
Setting E-Mail Notification Settings ...................................... 97
Entering Server/Database Connection Settings .......................... 98

Migrating to Windows Installer ........................................ 99
Repackaging a Sample Package .......................................... 100
Importing Packages into the Application Catalog .......................................... 102
Testing a Repackaged Application and Resolving Issues .......................................... 104
Distributing a Repackaged Application ...................................... 107

Migrating to Application Virtualization ...................................... 108
Identifying Packages to Virtualize .......................................... 109
Converting to Virtual Formats .......................................... 110
Testing and Distributing Converted Packages ...................................... 112

Testing for Application Compatibility ...................................... 114
Importing Packages, Web Applications, and Mobile Apps .......................................... 115
Selecting Tests to Run and Setting Default Fix Option .......................................... 116
Performing Testing and Viewing Results ...................................... 117
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

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting Started</td>
<td>Explains what is included in this Evaluation Guide and how to get started.</td>
</tr>
<tr>
<td>New Features in AdminStudio 2016 SP2</td>
<td>Provides an overview of the new features in AdminStudio 2016 SP2.</td>
</tr>
<tr>
<td>New Features in AdminStudio 2016 SP1</td>
<td>Provides an overview of the new features in AdminStudio 2016 SP1.</td>
</tr>
</tbody>
</table>
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](image)

**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>
<thead>
<tr>
<th>Table 2 • Evaluation Exercises</th>
</tr>
</thead>
<tbody>
<tr>
<td>Procedure</td>
</tr>
<tr>
<td>--------------------------------</td>
</tr>
<tr>
<td>Setting Up AdminStudio Infrastructure</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Migrating to Windows Installer</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Procedure</td>
</tr>
<tr>
<td>---------------------------------</td>
</tr>
<tr>
<td><strong>Migrating to Application Virtualization</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Testing for Application Compatibility</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
These four main procedures are also featured on AdminStudio's Start Page.

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:

![Migrate to Windows Installer Flowchart](image)

**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 PowerShell 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.

Figure 5: PowerShell Wrap Options Tab of Options Dialog Box
The **PowerShell Wrap Options** tab includes the following properties:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Template Directory</strong></td>
<td>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 <a href="#">PowerShell Script Template File</a>.</td>
</tr>
<tr>
<td><strong>Output Directory</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Wrap on Import</strong></td>
<td>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.</td>
</tr>
</tbody>
</table>
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\n```

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

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:

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

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).

![Import Wizard](image)

**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 PowerShell 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.

![Catalog Deployment Type](image)

**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.

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.

![Image](image.png)

**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.

![Image](image.png)

**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:

Figure 13: Banner Displayed in PowerShell Installer
You can customize this banner by replacing the content of the AppDeployToolkitBanner.png file.

![Custom Banner](image)

**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!

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

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

**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.

**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.

**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.

**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.

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:

![Figure 23: JRE Versions Installed by Packages](image)

**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.

![Figure 24: Best Practices and Risk Assessment Test Groups on the Select Tests to Execute Dialog Box](image)

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.

![Windows Server 2016 OS Compatibility Tests](image)

**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.

![Figure 26: Adding a New Group on Destination Group Panel of Distribution Wizard](image)

*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.

![Figure 27: Choosing a Connection on the Microsoft Configuration Manager Deployments Report](image)
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.

![Select Temporary Location Option](image)

**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.

![Microsoft UWP App Package (.appx) Option on Package Type Selection Panel of Import Wizard](image)

**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:

  ```powershell
  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:

  ```powershell
  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).

![Mac Operating System Application Compatibility](image)

**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.

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.

![Mac Installer Best Practices Report](image)

**Figure 32:** Mac Installer Best Practices Report
You can click on the bars in this report to drill down to view more detailed reports.

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.

![ACE Tests Tab of Application Manager Options Dialog Box](image)

**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.

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.

![Bundled Packages Tab of Catalog Deployment Type View for Apple Installer Package (.pkg)](image)

**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.

![Figure 37: New PKG Installer Choices Tab of Catalog Deployment Type View for Mac PKG Installer](image)

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:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Visible</strong></td>
<td>This option can be either selected or not selected:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Selected</strong>—This choice setting will be displayed in the installer.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Not selected</strong>—This choice setting will not be displayed in the installer.</td>
</tr>
<tr>
<td><strong>Selected</strong></td>
<td>This option can be either selected or not selected:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Selected</strong>—If this choice setting is displayed in the installer, its check box will be checked.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Not selected</strong>—If this choice setting is displayed in the installer, its check box will not be checked.</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>This option can be either selected or not selected:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Selected</strong>—If this choice setting is displayed in the installer, it will be enabled.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Not selected</strong>—If this choice setting is displayed in the installer, it will be disabled.</td>
</tr>
<tr>
<td><strong>Custom Location</strong></td>
<td>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.</td>
</tr>
</tbody>
</table>

*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)

![Figure 38: Apple Mac App Store Option on Public Store Selection Panel](image)

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.

![Figure 39: Genre(s) Field on Package Information Tab](image)

**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.

![Figure 40: Test Results for Mac OS X Desktop App](image)
In AdminStudio 2016, the following new tests for Mac OS X desktop applications have been added:

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

**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.

![Figure 41: Consolidated Test Results for Apple Installer Package (.pkg)](image)
When you view detailed test results, the name of the child .dmg, .pkg, or .app file that generated the error or warning is listed.

![Consolidated Test Results](image)

**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.

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.

![Figure 44: Selecting App Portal Options on Categories Dialog Box](image)

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.

![Figure 45: Catalog Item Field on App Portal Information Tab](image)
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.

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:

<table>
<thead>
<tr>
<th>Subtab</th>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Server Name</td>
<td>Name of the Casper server.</td>
</tr>
<tr>
<td></td>
<td>Category</td>
<td>Category in Casper that the package will be added to.</td>
</tr>
<tr>
<td></td>
<td>Note</td>
<td>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.</td>
</tr>
<tr>
<td>Info</td>
<td>Info</td>
<td>Information to display to the administrator when the package is deployed or uninstalled.</td>
</tr>
<tr>
<td>Notes</td>
<td>Notes</td>
<td>Notes to display about the package (such as the name of the person who built it and when it was built).</td>
</tr>
<tr>
<td></td>
<td>Note</td>
<td>Not displayed for Mac App Store apps.</td>
</tr>
<tr>
<td>Free</td>
<td>Free</td>
<td>Indicates whether or not the Mac App Store app is available for free (<strong>True</strong>) or whether it requires payment (<strong>False</strong>).</td>
</tr>
<tr>
<td></td>
<td>Note</td>
<td>Only displayed for Mac App Store apps.</td>
</tr>
<tr>
<td>Subtab</td>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Options</td>
<td>Priority</td>
<td>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.</td>
</tr>
<tr>
<td>Fill user templates</td>
<td></td>
<td>Set this property to True to fill new home directories with the contents of the home directory in the package's Users folder.</td>
</tr>
<tr>
<td>(FUT)</td>
<td></td>
<td>This setting can be changed when deploying or uninstalling the package using a policy.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong> • Only applicable to DMG packages.</td>
</tr>
<tr>
<td>Fill existing user</td>
<td></td>
<td>Set this property to True to fill existing home directories with the contents of the home directory in the package's Users folder.</td>
</tr>
<tr>
<td>home directories</td>
<td></td>
<td>This setting can be changed when deploying or uninstalling the package using a policy.</td>
</tr>
<tr>
<td>(FEU)</td>
<td></td>
<td><strong>Note</strong> • Only applicable to DMG packages.</td>
</tr>
<tr>
<td>Requires restart</td>
<td></td>
<td>Set this property to True to require that computers must be restarted after installing the package.</td>
</tr>
<tr>
<td>Install on boot drive</td>
<td></td>
<td>Set this property to True to ensure that the package is installed on the boot drive after imaging.</td>
</tr>
<tr>
<td>after imaging</td>
<td></td>
<td><strong>Note</strong> • 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.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Subtab</th>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limitations</td>
<td>Operating System Requirement</td>
<td>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</td>
</tr>
<tr>
<td>Install Only if Available in Software Update</td>
<td>Enable Only if Available</td>
<td>Set to True to require that this package only be installed if it is available in a software update.</td>
</tr>
<tr>
<td>Limit Architecture Type</td>
<td>True</td>
<td>Set to True to require that this package only be installed on machines matching the selected Architecture Type Requirement.</td>
</tr>
<tr>
<td>Architecture Type Requirement</td>
<td>True</td>
<td>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</td>
</tr>
<tr>
<td>Substitute Package</td>
<td>Substitute Package</td>
<td>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.</td>
</tr>
</tbody>
</table>

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.

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.

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.

![Figure 49: Consolidated Test Results for Windows Installer Suite Package](image-url)

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.

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

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.

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.

4. Enter the following information:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product Name</strong></td>
<td>The basic name of the application, excluding references to versions or editions, and without mentioning the publisher.</td>
</tr>
<tr>
<td><strong>Version</strong></td>
<td>The release number (or release identifier) of an application.</td>
</tr>
<tr>
<td><strong>Edition</strong></td>
<td>Enter the edition of this application.</td>
</tr>
<tr>
<td><strong>Publisher</strong></td>
<td>The name of the publisher of this software, responsible for its development and distribution.</td>
</tr>
</tbody>
</table>
5. Click **Create**. A confirmation message appears stating that a new local Flexera Identifier has been created.

**Classification**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification</td>
<td>To indicate how this application is classified, select one of the following options:</td>
</tr>
<tr>
<td>• Beta</td>
<td>A pre-release application (covers such items as beta releases, alpha releases, or release candidates) that you have under some special arrangement.</td>
</tr>
<tr>
<td>• Commercial</td>
<td>The application requires a license to be purchased for use in a commercial setting.</td>
</tr>
<tr>
<td>• Freeware</td>
<td>Licensed for use in a commercial environment free-of-charge.</td>
</tr>
<tr>
<td>• Malware</td>
<td>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.</td>
</tr>
<tr>
<td>• Shareware</td>
<td>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.</td>
</tr>
<tr>
<td>• X Rated</td>
<td>The application contains potentially objectionable or sexually explicit material. You might want to consider whether corporate policies require any action.</td>
</tr>
<tr>
<td>• Update</td>
<td>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).</td>
</tr>
</tbody>
</table>
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.

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

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 Automatics 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.

![Figure 52: Automatically Execute Tests After Import Option](image)

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.

![Figure 53: Import Options > Package Auto Import Tab of Options Dialog Box](image)

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.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directory</td>
<td>Enter the directory to monitor. You can either enter a directory path or click the browse button and select a directory.</td>
</tr>
<tr>
<td></td>
<td>Important • The directory must be in UNC format and it must be a shared directory.</td>
</tr>
<tr>
<td>Target Group</td>
<td>Specify the group in the Application Manager tree into which the packages will be imported.</td>
</tr>
<tr>
<td>Monitoring Status</td>
<td>Set this option to True to begin monitoring this directory for new and updated packages. Set it to False to stop monitoring the directory.</td>
</tr>
<tr>
<td>Extensions</td>
<td>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.</td>
</tr>
</tbody>
</table>

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.

![Monitoring the Import of Packages in a Directory on the Package Auto Import Tab](image)

**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.

![Select Watcher Extensions Dialog Box](image)

**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.

![Select Applications Panel of Import Wizard](image)

**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:

![Edit Settings Button on Plugin Options > Automated Application Converter Plugin Tab](image)

**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.

![Editing the Default Automated Application Converter Settings File](image)

**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 the 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.

![Automated Application Converter Settings Panel](image)

**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.

![Figure 60: Editing the Default Conversion Settings](image)

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.

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

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.
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.

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.

![Create Transform with InstallShield](image)

**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.

**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.

![New "Open File Location" Command on Shortcut Menu](image)

**Figure 64:** New "Open File Location" Command on Shortcut Menu

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

![File Location of Imported Package's Source Files](image)

**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.

![SCCM Settings Tab of Application View](image)

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.

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

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Create an Application Catalog</td>
<td>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.</td>
<td>AdminStudio is open and connected to a new Application Catalog named AdminStudio Evaluation Catalog.</td>
</tr>
</tbody>
</table>

Note • On the Select Software Repository Location panel of the Application Catalog Wizard, do NOT select the Enable Software Repository option.

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**

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Run Virtual Machine Preparation setup to enable auto login</td>
<td>On a Microsoft Hyper-V Server image, VMware ESX/ESXi Server image, or VMware Workstation (6.5+) image, run the virtual machine preparation setup.</td>
<td>When you restart the virtual machine image, you are automatically logged in and GuestAgent.exe is launched.</td>
</tr>
<tr>
<td></td>
<td>Note • For instructions, see Preparing Your Virtual Machines for Use With the Automated Application Converter.</td>
<td><img src="image" alt="Image of Automated Application Converter - Guest Agent" /></td>
<td>When you restart the virtual machine image, you are automatically logged in and GuestAgent.exe is launched.</td>
</tr>
<tr>
<td>2</td>
<td>Create a snapshot for repackaging</td>
<td>On the prepared virtual image, create a clean snapshot named AutoRepack_Base.</td>
<td>A clean snapshot named AutoRepack_Base exists on the virtual machine.</td>
</tr>
<tr>
<td></td>
<td>Note • For instructions, see Taking a Snapshot in Preparing Your Virtual Machines for Use With the Automated Application Converter.</td>
<td><img src="image" alt="Image of Automated Application Converter - Guest Agent" /></td>
<td>A clean snapshot named AutoRepack_Base exists on the virtual machine.</td>
</tr>
</tbody>
</table>
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.

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:
  

**Note**  
For instructions, see **VMware VIX API Requirement**.
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.

---

### Table 4 • Configure Virtual Machines

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.</td>
<td>Add a virtual machine to Automated Application Converter</td>
<td>Open Automated Application Converter by clicking <strong>Add Virtual Machines</strong> on the <strong>Set Up Infrastructure</strong> tab of the AdminStudio Start page. The <strong>Application Conversion Project Wizard</strong> opens. Cancel the wizard. Then open the <strong>Machines</strong> tab and follow the instructions in <strong>Adding Virtual Machines Using the Virtual Machine Import Wizard</strong> to add the prepared virtual machine to the <strong>Machines</strong> tab.</td>
<td>A virtual machine is listed on the <strong>Machines</strong> tab of Automated Application Converter.</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Set SMTP Notification Settings</td>
<td>On the Notification Settings tab of the AdminStudio Options dialog box, enter your SMTP settings for e-mail notifications.</td>
<td>When you click Test on the Notifications Settings tab, a successful message opens.</td>
</tr>
</tbody>
</table>

*Note* • For instructions, see Setting E-Mail Notification Settings.
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

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enter System Center Configuration Manager connection settings</td>
<td>Open the Application Manager Options dialog box, and on the Distribution System tab, create a named connection to System Center 2012 Configuration Manager.</td>
<td>When you click Test on the Distribution System tab, the following message is displayed: Connection to ServerName Succeeded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note • For instructions, see Creating a New Distribution System Connection Setting</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Entering Microsoft ACT database connection settings</td>
<td>Open the Application Manager Options dialog box, and on the Microsoft ACT tab, enter Microsoft ACT database connection information.</td>
<td>When you click Test on the Microsoft ACT tab, a successful message opens: Connection to Microsoft ACT database successful</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note • For instructions, see Entering Microsoft ACT Database Connection Settings</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
<td></td>
</tr>
</tbody>
</table>
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.

**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 re packaging 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

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Select and repackage a sample package (.exe) to a Windows Installer Package (.msi)</td>
<td>Use Repackaging Wizard (Installation Monitoring method) to repackage the following sample setup: C:\AS2016EvalGuideDataFiles\SampleApplicationSetup\SampleApplicationSetup.exe</td>
<td>The captured data was converted into a Repackager project file (SampleApplication.irp) and opened in the Repackager interface.</td>
</tr>
<tr>
<td>2.</td>
<td>Important • It is recommended that you repackage this sample package on a clean machine. Save the captured data in the following directory: C:\Packages</td>
<td>Note • For instructions, see Repackaging Using the Installation Monitoring Method.</td>
<td></td>
</tr>
</tbody>
</table>
### Table 7 • Repackage a Sample Package

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.</td>
<td>Edit package in Repackager and build Windows Installer package</td>
<td>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.</td>
<td>The [WindowsVolume] folder is displayed in red to indicate that it is excluded:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>![folder_icon] ![folder_icon] ![folder_icon]</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note •</strong> For instructions, see Excluding All Files in a Directory.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Open the Repackaged Output view and click Build to build a Windows Installer package.</td>
<td>The Repackager project file has been converted to a Windows Installer package, located in the following directory:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C:\Packages\MSI_Package\SampleApplication.msi</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note •</strong> For instructions, see Building a Windows Installer Package.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>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:</td>
<td>New Windows Installer package (and other associated files) are now in the same main directory as the other evaluation data files.</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.</td>
<td>Import Windows Installer package into Application Catalog</td>
<td>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.</td>
<td>The packages are now listed in the Application Manager tree, each under its own Application node:</td>
</tr>
</tbody>
</table>

Note • For instructions on how to import a directory of packages into the Application Catalog, see Importing a Folder of Multiple Packages.
### Table 8 • Import Into Application Catalog

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.</td>
<td>Import Windows Installer package into Application Catalog</td>
<td>Create a new group in the Application Manager tree named Engineering and move the <strong>Sample Application</strong> into that new group.</td>
<td>The Application Manager tree should now be organized as follows:</td>
</tr>
</tbody>
</table>

**Note** • For instructions, see Adding Groups and Organizing Applications in Application Manager.

Create another new group in the Application Manager tree named Marketing, and move the other three applications into that group.

The Application Manager tree should now be organized as follows:
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

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.</td>
<td>Perform Windows Installer best practices and OS compatibility testing</td>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

**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:

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

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.</td>
<td>Perform application conflict testing</td>
<td>Use the Conflict Wizard to detect conflicts between the following two packages:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Source: Sample Application Source</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Target: Sample Application Target</td>
<td></td>
</tr>
</tbody>
</table>

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:

![Test Center Deployment Type View](image)

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
8. **Perform automatic issue resolution**  

Issues for which automatic fixes are available are identified by the Error With Fix or Warning With Fix icon:

![Error With Fix icon](image)

![Warning With Fix icon](image)

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

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.</td>
<td>Distribute repackaged application</td>
<td>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.</td>
<td>SampleApplication.msi is copied to the specified network location, making it available to your enterprise.</td>
</tr>
</tbody>
</table>

**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.*
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.

![Flowchart](image)

**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

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Import packages into Application Catalog.</td>
<td>Open Application Manager and locate the packages that you imported in Importing Packages into the Application Catalog.</td>
<td>The four Windows Installer packages are listed in the Application Manager tree:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><img src="image1.png" alt="Application Manager screenshot" /></td>
</tr>
<tr>
<td>2.</td>
<td>View package’s Application Virtualization Compatibility test results.</td>
<td>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.</td>
<td>The packages’ application virtualization compatibility test results are listed in the Application Virtualization Compatibility column of the Test Center Group View.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><img src="image2.png" alt="Test Center Group View screenshot" /></td>
</tr>
</tbody>
</table>
Table 11 • Identify Packages to Virtualize

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Identify candidates for virtualization.</td>
<td>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.</td>
<td>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. However, since 64-bit packages are supported by Microsoft App-V, this package is ready for conversion to App-V format.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Import candidate packages into Automated Application Converter.</td>
<td>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.</td>
<td>Packages are listed on the Packages tab.</td>
</tr>
</tbody>
</table>
### Table 12 • Convert to Virtual Formats

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Convert to virtual packages.</td>
<td>First, open the <strong>Project Options</strong> dialog box and make sure that the <strong>Package Creation</strong> property is set to <strong>App-V 4.6 with AdminStudio</strong>. 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. <strong>Important</strong> • 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.</td>
<td><img src="image" alt="Package Structure Diagram" /></td>
</tr>
<tr>
<td>6</td>
<td>Test launch virtual packages.</td>
<td>Test the virtual packages by launching them on a virtual machine. Virtual packages launch successfully. Note • For instructions, see Performing Automated Testing of App-V Packages.</td>
<td><img src="image" alt="Application Catalog Tree" /></td>
</tr>
<tr>
<td>7</td>
<td>Publish virtual packages to Application Catalog.</td>
<td>Import the new virtual packages into the Application Catalog. The virtual packages are listed in the Application Manager tree under their associated Application. Note • You may need to click Ctrl + R to refresh the Application Catalog tree.</td>
<td><img src="image" alt="Application Catalog Tree" /></td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.</td>
<td>Perform virtualization best practices testing.</td>
<td>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.</td>
<td>When testing is complete, results are displayed on the Summary tab of the Test Center Deployment Type View.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>Perform conflict testing.</td>
<td>In Application Manager, perform conflict testing of the Sample Application Source App-V package against the Sample Application Target App-V package.</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Table 13 • Test and Distribute Converted Packages

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Edit App-V packages (if necessary).</td>
<td>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.</td>
<td>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.</td>
</tr>
<tr>
<td>11</td>
<td>Distribute to enterprise for user acceptance testing and production.</td>
<td>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.</td>
<td>The selected App-V package is copied to the specified network location, making it available to your enterprise.</td>
</tr>
</tbody>
</table>
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.

![Figure 70: Test for Application Compatibility Tab of AdminStudio Start Page](image)

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. | The Application Manager tree should now be organized as follows:          |
|        | **Note** • For instructions on how to import a directory of packages into the Application Catalog, see Importing a Folder of Multiple Packages. |                                                                                               | [Diagram of Application Manager tree]                                     |
| 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. | The web applications are listed in the Application Manager tree:          |
|        | **Orbitz** at: http://www.orbitz.com | [Diagram of Application Manager tree with imported web applications: Orbitz, Travelocity]    | [Diagram of Application Manager tree with imported web applications: Orbitz, Travelocity] |
|        | **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. |                                                                                   |
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

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Select the operating system and browser compatibility tests that you want to run.</td>
<td>Select the <strong>Operating System Compatibility</strong> and <strong>Browser Compatibility</strong> tests that you want to run, as described in Selecting Tests to Execute.</td>
<td>The <strong>Operating System Compatibility</strong> and <strong>Browser Compatibility</strong> tests that you want to run are selected on the Select Tests to Execute dialog box.</td>
</tr>
</tbody>
</table>

*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.
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**

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.</td>
<td><strong>Click Execute Tests to test Windows Installer packages, mobile apps, and web applications (statically).</strong></td>
<td>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.</td>
<td>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.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Result</th>
</tr>
</thead>
</table>
| 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. |
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.

Table 16 • Performing Testing and Viewing Results

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.</td>
<td><strong>Click Launch Web Test to test web applications interactively.</strong></td>
<td>To interactively, dynamically test a web application, select a web application node in the tree and click <strong>Launch Web Test</strong>.</td>
<td>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).</td>
</tr>
</tbody>
</table>
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:

![Operating System Compatibility Test Results Table]

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

![Operating System Compatibility Detailed Test Results]

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

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Instructions</th>
<th>Result</th>
</tr>
</thead>
</table>
| 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. | 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. |

*Note • For more information, see Performing Automatic Issue Resolution.*