

SCCM Application Packaging

Part 1 – The Process & SCCM Interface

A ZEN GUIDE TO SCCM APPLICATION PACKAGING WITH POWERSHELL

JERRY "ZEN" SENFF

This document is for informational purposes only.

NO WARRANTIES, EXPRESS OR IMPLIED, ARE MADE AS TO THE INFORMATION IN THIS DOCUMENT.

Copyright © 2015 Jerry Senff. All rights reserved.

Document updates copyright © 2019 Jerry Senff. All rights reserved.

Microsoft, Windows, MSDN, and System Center Configuration Manager are trademarks or registered trademarks of Microsoft Corporation in the USA and other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Adjusting Token Privileges in PowerShell © 2010 Precision Computing. Retrieved from <http://www.leeholmes.com/blog/2010/09/24/adjusting-token-privileges-in-powershell/>

Contents

Documentation	4
Potential Packaging Candidates	5
Packaging Process	6
Shared Components	7
Blocked Applications	7
Required Packaging Information	8
Information Required to Successfully Package an Application for SCCM Distribution	8
Application Package Licensing	8
General Rules	9
Documentation Produced	9
Packaging Template	10
Template	10
Status Email	11
Status Email Sample	11
How to Package Applications	12
1. How to Obtain the Package Details	13
2. How to Package MSI Installers	18
3. How to Package Script Installers	29
4. How to Distribute Content	53
5. How to Deploy a Package	56
6. Testing the Application Package	60
7. Re-Testing the SCCM Application Package	64
Updating Content on the Distribution Server	64
Verifying the Content Update on the Distribution Server	65
Clearing the Configuration Manager Cache	67
Program Visibility Options	69

Documentation

The documentation has been broken into three components to make both reading and downloading from GitHub easier:

- Part 1 covers the application packaging process and the SCCM interface.
- Part 2 covers the PowerShell programming tips and tricks used with SCCM.
- Part 3 is a collection of example PowerShell install and uninstall programs demonstrating the PowerShell tips and tricks used with SCCM for the successful installation of complex software packages.

Potential Packaging Candidates

Every software application is a potential candidate for Microsoft System Center Configuration Manager (SCCM) distribution. Often, the only way to know if the application can be packaged is to install the application and see what happens. Applications can install, but not run on Microsoft Windows 7, or they may install and run, but not uninstall.

At the contract start, a local business had around 65 applications in their SCCM software library. Of the 65 existing applications, only 15 worked correctly in that the applications could uninstall successfully through SCCM. Most of the 65 applications did not have a SCCM uninstall method in place or had bad detection methods preventing SCCM from knowing the application even existed on a client system. Now, each of the 598 SCCM application packages successfully installs, detects the application's existence, and uninstalls the application along with all components.

This guide covers the application packaging process developed over a ten-month period of packaging applications daily. The target audience is IT Professionals with little or no prior System Center Configuration Manager, application packaging, or PowerShell experience. Screenshots are included in the how-to package section to provide clarity. The PowerShell examples are written simply in a modular fashion so snippets may be copied from the document into a PowerShell ISE window to easily build a PowerShell install or uninstall program. All the PowerShell [Start-Process](#) commands are executed with the `-wait` switch since Windows only allows one application to install at a time, though the application installing may call other installers during its setup process. An effort was made to make the program execution window output look uncluttered so technicians could more easily track the progress of the application installation. The process presented in this guide worked successfully for packaging over 550 software applications.

Packaging Process

Every application is installed a minimum of three times during the packaging process:

- First time, manually install the application from a **Command Prompt (Admin)** window:
 - Execute the setup program in the command window with a `/?` (or `-?`) to see if there are any switches available supporting silent installation.
 - Note any help dialog that appears and take a screenshot for the OneNote document.
 - Open a browser and search using the application name followed by 'silent install' to see if anybody else has had any success silently installing the application.
 - Basic questions to ask:
 - Does the application install?
 - Does the application run?
 - Are shortcuts created on the **Desktop** or in the **Start Menu** for the user?
 - Documenting the application:
 - Create a OneNote document (or preferred document software) using the packaging template.
 - Open the **Programs and Features** control panel applet and sort by the **Installed On** column:
 - Does the application register?
 - How many other components were installed with the application?
 - Open the registry editor (regedt32) and obtain the following information for the application and every component the application installs:
 - Uninstall registry key path
 - **UninstallString** value
 - **DisplayName** value
 - **DisplayVersion** value
 - **Publisher** value
 - Uninstall the application and any components:
 - If the UninstallString is an .EXE or the application installs extra components:
 - Create a PowerShell program to remove the application and components.
 - Execute the program to ensure it works correctly.
 - Check **Programs and Features** to ensure the program no longer displays.
 - Check to ensure the shortcuts created by the install get removed.
 - Check for leftover installation folders.
 - If the **UninstallString** is a .MSI, copy the string to the Administrator command window and execute:
 - Check **Programs and Features** to ensure the program no longer displays.
 - Check to ensure the shortcuts created by the install get removed.
 - Check for leftover installation folders.
- Second time, attempt to install the application silently:
 - Are shortcuts created on the **Desktop** or **Start Menu** for the user?
 - Do all components install correctly?
 - Does the application run?
 - Does the application uninstall silently?
 - If there is a PowerShell program for uninstalling the application, check the program again.
- Package, distribute, and deploy the application using the SCCM Console.
- Third time, test the SCCM application package through the **Software Manager** interface on the application packaging workstation:
 - Are shortcuts created on the **Desktop** or **Start Menu** for the user?

- Do all components install correctly?
- Does the application run?
- Does the application uninstall?
- Are the shortcuts removed?
- Are the installation folders removed?

Shared Components

Many applications require certain shared components to be installed for the application to execute correctly. Shared components include redistributables and runtime libraries. Ideally, it is best to package any shared redistributables and runtimes for deployment through SCCM and to use an install dependency in the application package to install the shared component. Often, an application setup will install a specific redistributable or library version, regardless of the existence of the newest version on the same system, which complicates managing shared components through SCCM.

The solution implemented at the local business has all the latest versions of shared redistributables and runtime libraries packaged for SCCM distribution. Older versions of the .NET Framework are also packaged. When an application setup installs a down-level version of a redistributable or runtime, the information is added to the *Application Dependency* tracking spreadsheet and the PowerShell uninstall program will prompt the technician on whether the redistributable or runtime should be uninstalled.

Blocked Applications

Applications can be blocked from packaging for a variety of reasons. The number one reason at the local business was a lack of available setup files for various applications.

Blocked applications breakdown:

- No setup files available: 165 apps (73%)
- Windows 7 incompatibility: 32 apps (14%)
- No license key available: 7 apps (3%)
- SCCM specific issues: 3 apps (1%)
- Required CD-ROM: 3 apps (1%)
- Installer Issues: 4 apps (2%)
- Blocked by Group Policy: 6 apps (3%)
- Miscellaneous: 8 apps (3%)

Required Packaging Information

Information Required to Successfully Package an Application for SCCM Distribution

- The location of setup files for the application:
 - If obtaining from a web download, include a website link.
 - If a user account is required to download the application, an onsite client employee should obtain the setup files rather than the technician.
 - If obtaining from a network share, read/write permissions will be required to copy the setup files to the SCCM **Software Vault** share for packaging.
- Any information or special instructions regarding the following:
 - License keys
 - User name
 - Company name
 - File/folder copying
 - Shortcuts
 - Network share connections
 - Database connections
 - Ports
 - Authentication types
 - File/folder ACL changes
 - Security group assignments
 - Site servers
 - Any other items requiring additional PowerShell program development time to install and/or uninstall
- Access to network shares and/or databases will be needed not only to successfully install the program to create the SCCM package, but for any technicians that will be installing the package on an end-user's system.
 - A screenshot of the relevant registry key containing the connection information can also be used to create the key and settings directly with PowerShell.

Application Package Licensing

- If the application is a single license package, the license key will not be included with the SCCM package but will be required to test the SCCM package to ensure the package installs correctly.
- If the application has a site, group, or enterprise license, the license will be included with the SCCM package unless otherwise specified. This normally requires using the software publisher's tool or methodology to embed the license key in the application's setup files.
- Installs requiring license keys to complete setup and/or user names and passwords to access any resources will be packaged with "- **Manual**" in the package name, and a technician will be required to manually enter the information during the setup process.

General Rules

- If the application cannot be installed successfully on the desktop test system, it cannot be packaged for SCCM distribution.
- Not all applications can be automated to install without user interaction.
- If an application installs multiple software components, all components will be uninstalled using a PowerShell program requiring additional development time.
 - The technician uninstalling the application will be prompted on whether to uninstall shared software components such as Microsoft Visual C++ Redistributables and runtime libraries.
- Each environmental configuration change made during an installation using PowerShell will need to be reversed during an uninstall using PowerShell, which requires additional development time.

Documentation Produced

Application Packaging tracks and documents the following items:

- OneNote document (or preferred document software) created in the Packaging section for every packaged application:
 - See the "Packaging Template" document below for an example.
- OneNote document (or preferred document software) created in the Reference section for every PowerShell program:
 - Install programs
 - Uninstall programs
 - Detection methods
- SCCM packaged applications inventory spreadsheet with applications sorted by name and categories (e.g., redistributables, multimedia, research, productivity, etc.).
- Application packaging software dependencies between various applications to identify which items should prompt the technician during an uninstall. Examples:
 - Adobe Acrobat
 - Adobe Flash Player
 - Apple Bonjour
 - Microsoft .NET Frameworks
 - Microsoft Product Runtime releases
 - Microsoft Visual C++ Redistributables
- Blocked application examples:
 - No setup files available
 - Incompatible with Windows 7
 - Installs but does not start
 - Installs but does not uninstall
 - Needs newer version
 - No license key available to complete setup
 - Requires CD to run
- Applications on hold that should not be packaged until authorized:
 - SQL Server releases blocked by Group Policy
 - VPN Client

Packaging Template

Every application packaged for SCCM distribution will have a corresponding OneNote document (or preferred document software) in the Packaging section containing the details required to successfully package each application in case the existing package is corrupted or deleted and needs to be repackaged quickly.

Template

<p>Application Name</p> <p><u>Description</u> Brief product description.</p> <p><u>Package Notes/Comments</u> – Copy to all Notes/Comments sections in SCCM creating package. Free software license. Single license package. Group/Enterprise/Site license. Any specific instructions for installing the application including estimated time for large installs.</p> <p>Source file location (share or weblink):</p> <p>Copy source files to Software Vault share location:</p> <p>Import <i>appName.exe/.msi</i> into SCCM Client Applications as a <i>Script/MSI Installer</i>.</p> <p>Installation program:</p> <p>Uninstall program:</p> <p>Detection Method:</p> <p>User Experience: Installation behavior: Logon requirement: Installation program visibility: Enforce specific behavior: No specific action</p> <p>Dependencies (list redistributables, runtime libraries, etc., available through SCCM):</p> <p>Notes GUID: {appNameGUID} (32-bit/64-bit)</p> <p>Registry key or file path (used to check if application is installed):</p> <p>UninstallString (determines uninstall method to use):</p> <p>DisplayName: DisplayVersion: Publisher:</p>
--

Note: Microsoft OneNote was already in use at this location for documentation. Any document software may be used. The key here is being consistent in creating a "recipe" for every application packaged, including any PowerShell programs used with the package.

Status Email

A status email is generated to the SCCM Application Packaging team every day, or less often when packaging is slow. The email contains the current total packages in SCCM along with a breakdown of new applications packaged, existing packages tested, and application packages pending. Links to the various spreadsheets that the SCCM Application Packager maintains are also included along with their current numbers. There are currently three spreadsheets and a OneNote that are updated daily as part of the application packaging process.

Status Email Sample

To: SCCM Project Manager
CC: SCCM Team Members, Other Stakeholders
Subject: App Packaging Status

Total packages = 585

Created documentation in OneNote Reference section for updating the *ComplianceTest.ps1* program to create new Surface Pro 3 driver packages along with driver name/version troubleshooting steps.

New apps packaged:

- Adobe Flash Player 16 ActiveX
- Adobe Flash Player 16 Plugin
- ProcartaPlex Analyst - Manual

Existing packaged apps tested:

- Adobe Flash Player 15 ActiveX
- Adobe Flash Player 15 Plugin

Pending apps:

- None

The ***SCCM Packaged Apps - Categories*** spreadsheet has been updated to include all applications currently in SCCM:

- Spreadsheet is located at:
 - \\server\path\SccmPackaging\SCCM Packaged Apps - Categories.xlsx

Blocked apps: Total = 238

- Blocked apps have been broken out into a separate spreadsheet located at:
 - \\server\path\SccmPackaging\BlockedApps.xlsx, ***Blocked Applications*** sheet

Apps on hold: Total = 8

- Apps on hold have been broken out into a separate spreadsheet located at:
 - \\server\path\SccmPackaging\BlockedApps.xlsx, ***Apps on Hold*** sheet

Application Dependencies:

- Application dependencies are listed in the following spreadsheet located at:
 - \\server\path\SccmPackaging\AppPckgDependencyMap.xlsx

How to Package Applications

To package applications for SCCM distribution, the following client site domain user accounts are required:

- Domain user account (or equivalent) to log on for testing published application packages:
 - Add to onsite physical system's **Local Administrators** security group
- Domain engineer user account (or equivalent):
 - Read/write permissions for **Software Vault** share
 - Read/write permissions for all shares with application setup files
 - SCCM Security Roles:
 - Application Administrator
 - Desktop Technician
 - SCCM Role-Based Collection to test deployments:
 - IS Desktop Test

1. How to Obtain the Package Details

Software required:

- 7-Zip (<http://www.7-zip.org/download.html>)
- Orca MSI Editor ([https://msdn.microsoft.com/en-us/library/windows/desktop/aa370557\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa370557(v=vs.85).aspx))
- regedt32.exe (Located in the %SystemRoot%/System32 folder)
- Microsoft OneNote (or preferred document software)

In the *Application Packaging Notes* document, create a new page under *SCCM Packaging* to document the application being packaged.

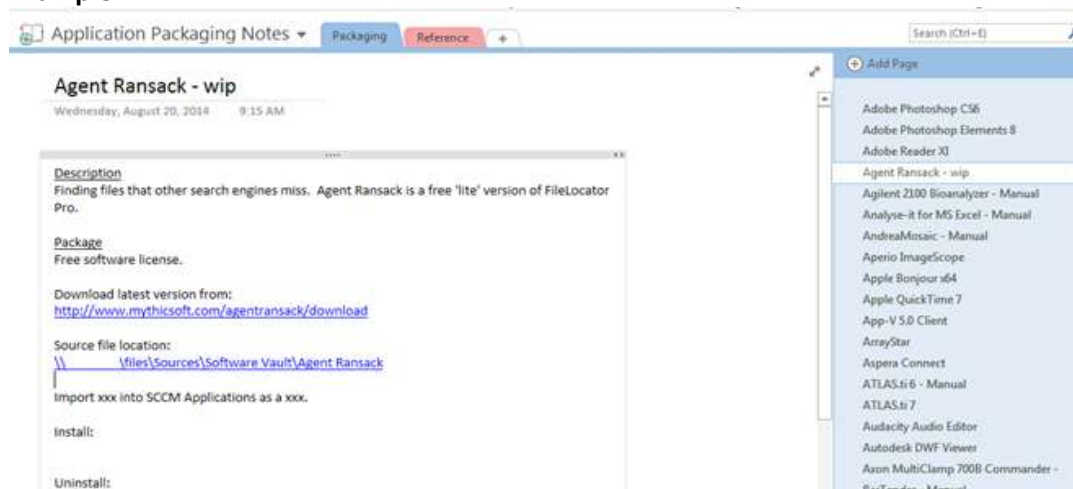
- Naming convention:
 - Applications not requiring user intervention: **ApplicationName**
 - Applications requiring user intervention: **ApplicationName - Manual**
 - Applications being packaged need a - **wip** after the name to indicate a "work in progress"
- Copy the contents of the **Packaging Template** template into the newly created page (see *Packaging Template* section above).
- Be sure to fill out all the information while progressing through the packaging process.

Locate the application install folder with the setup file or the webpage containing a download link.

In the application packaging document:

- Find a brief description of the application for the **Description** section at the top of the page.
- Determine the type of licensing for the application:
 - Free software license - No purchase required.
 - Single software license - Every install will need a unique key to finish the setup wizard or to activate the product after installation.
 - Enterprise/Group/Site license - The license key is included in the installer and the user does not need to provide a license.
- Document the source of the setup file.
 - Internal share: Create a link containing the path to the internal share.
 - External download: Create a link to the download webpage.

Example:



- Find the install folder with the setup file(s), or the download webpage, and copy/download the setup file(s) to a new folder in the SCCM **Software Vault** share.
- Open **Explorer** and browse to the install folder containing the setup file.
- Right-click the setup file and select **7-Zip, Open archive** to determine if the setup file is a self-extracting zip with an MSI or a setup.exe file that can be extracted to the install folder. Extract the files if possible to the install folder.
- If the install file is not a self-extracting zip file, open a **Command Prompt (Admin)** window.
- Using the copied folder path from above, type **pushd** and paste in the folder path to switch to the setup folder on the Software Vault share and press **Enter**.

```
pushd \\server\share\Software Vault\AppName
```

- Type in the name of the install file followed by a forward-slash and question mark to determine if the setup file has any setup options for a quiet install. -? can also be tried.

```
setup.exe /?
```

- If there are no setup options available, run the setup executable and stop at the setup wizard startup screen.
- Open Explorer and browse to the user **%temp%** folder and look for the setup files. For example:

```
C:\Users\UserName\AppData\Local\Temp\Mythicsoft\AgentRansack_7.0.822.1
```

- Copy all install folders in the **%temp%** folder to the install folder on the Software Vault share.
- Finish the setup wizard.

Start the application to make sure it installed correctly.

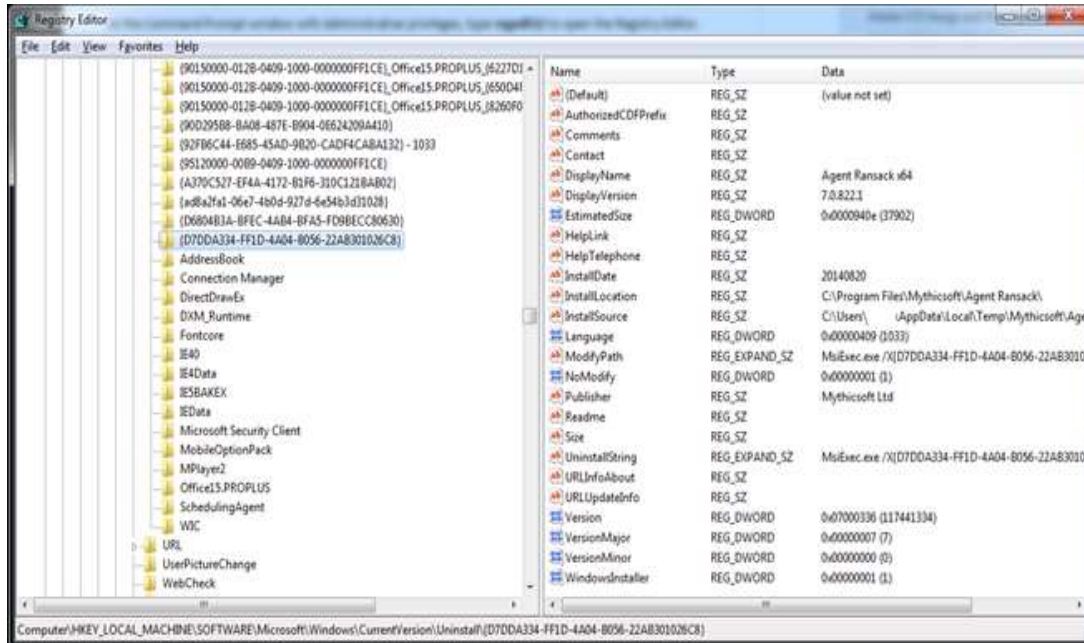
- A shortcut in the **Start Menu** or on the **Desktop** should be present and the application should open.
- Open the **Programs and Features** control panel applet and sort the **Installed On** column so the most recently installed programs are at the top.
- Make note of any newly installed applications like redistributables and runtime libraries.
- In the **Command Prompt(Admin)** window, type **regedt32** to open the Registry Editor.
- Find the Uninstall registry key for the application.
 - 64-bit applications will be found at:

```
HKLM: SOFTWARE\Microsoft\windows\CurrentVersion\Uninstall
```

- 32-bit applications will be found at:

```
HKLM: SOFTWARE\wow6432Node\Microsoft\windows\CurrentVersion\Uninstall
```

Agent Ransack Example:



In the OneNote document:

- Scroll to the bottom of the OneNote application package page to the **Notes** section.
- Copy the registry key path and paste under **Registry key**.
- Copy the application GUID and paste after **GUID:** at the top of the **Notes** section and note whether the GUID is 32-bit or 64-bit.
- Copy the data from the **UninstallString** parameter value and paste under **UninstallString:**
- Copy the data from the **DisplayName** parameter and paste after **DisplayName:**
- Copy the data from the **DisplayVersion** parameter and paste after **DisplayVersion:**
- Copy the data from the **Publisher** parameter and paste after **Publisher:**

Example:

Notes

GUID: {D7DDA334-FF1D-4A04-B056-22AB301026C8} (64-bit)

Registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{D7DDA334-FF1D-4A04-B056-22AB301026C8}

UninstallString:

MsiExec.exe /X{D7DDA334-FF1D-4A04-B056-22AB301026C8}

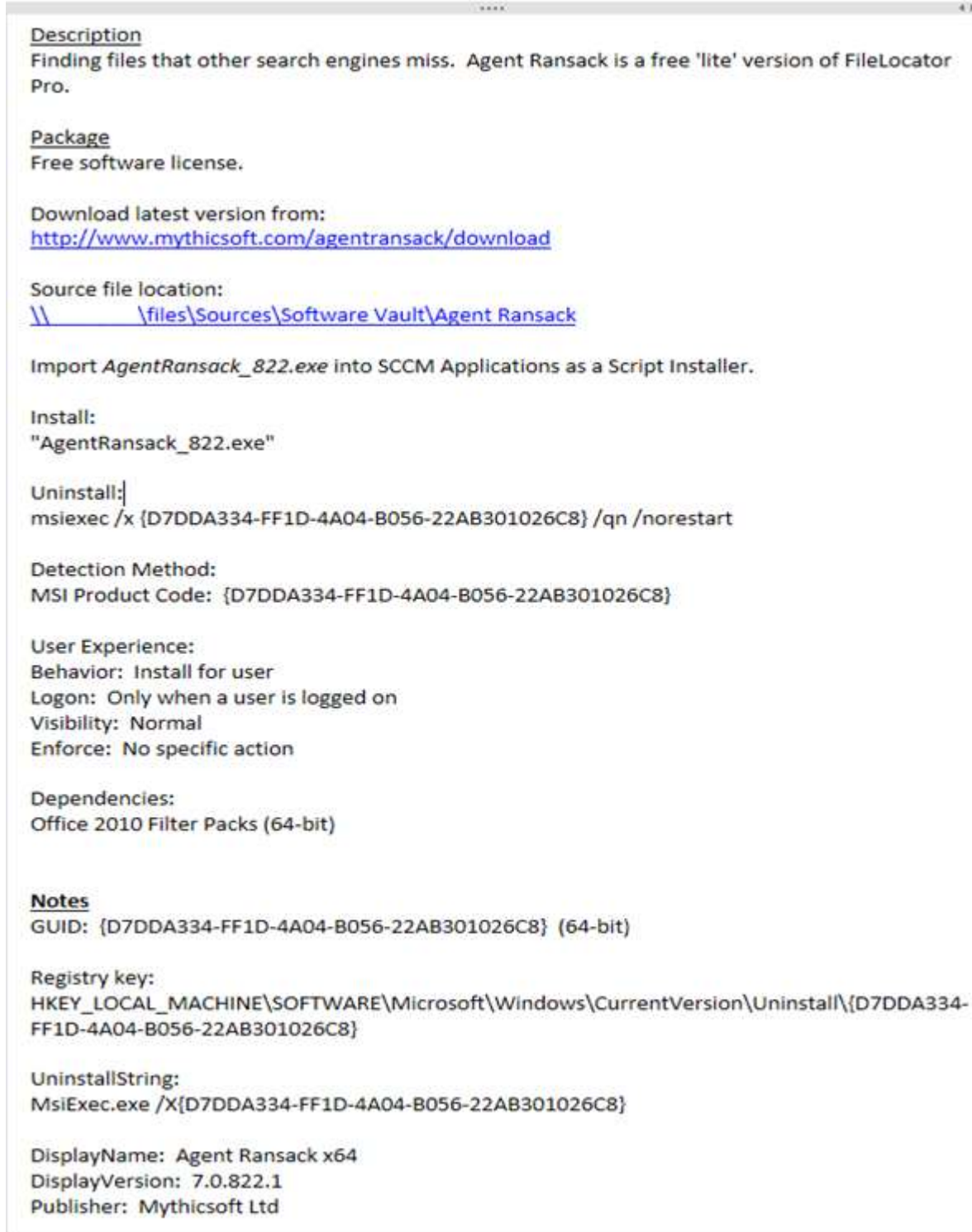
DisplayName: Agent Ransack x64

DisplayVersion: 7.0.822.1

Publisher: Mythicsoft Ltd

If the **UninstallString** is an MsiExec call like above, you now have all the information needed to package the application.

The completed OneNote document for the Agent Ransack application package:



If the **UninstallString** calls an uninstall program, which may or may not take parameters, a PowerShell program may be required to uninstall the application. See the **SCCM Application Packaging - Part 2** -

PowerShell.docx documentation for more information about PowerShell program examples, snippets, detection methods, and install/uninstall methods.

Example:

UninstallString:

C:\Program Files (x86)\InstallShield Installation Information\{F60B80F1-7F44-4491-AD8D-7100A3F66A44}\setup.exe -runfromtemp -l0x0409

Uninstall the application by pasting the **UninstallString** into the **Command Prompt (Admin)** window and press **Enter** to uninstall the application. Check the following after the application uninstalls:

- Open the **Programs and Features** control panel applet to ensure that the application is no longer listed.
- Check the registry to ensure that the application **Uninstall** key was removed during the uninstall process.
- Open **Explorer** and browse to the application install folder to see if any leftover files or folders remain.
- Open the **Start Menu** and ensure that the application shortcuts were removed. Any **Desktop** shortcuts should also be removed.

Move on to the **2. How to Package MSI Installers** section or the **3. How to Package Script Installers** section.

2. How to Package MSI Installers

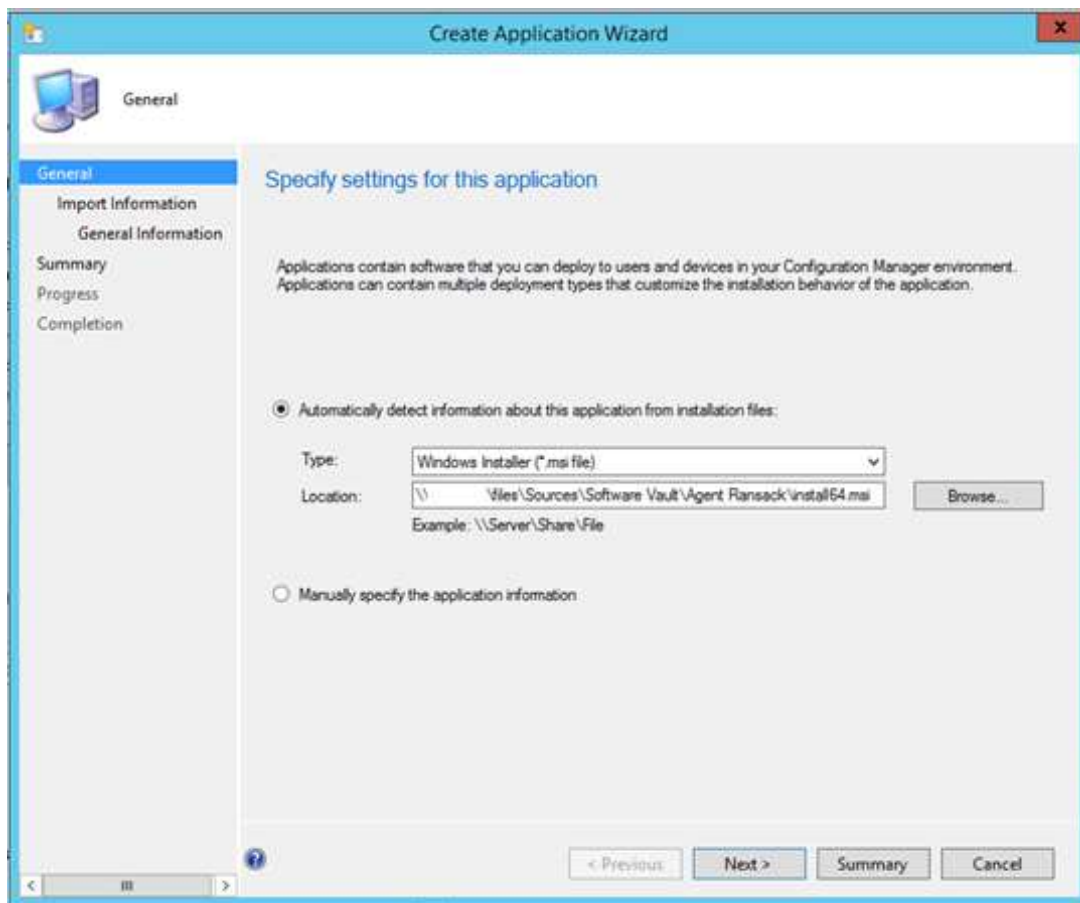
To create the MSI Installer:

Open a Remote Desktop session to the server running SCCM and logon using a user account with appropriate permissions.

Start the **System Center 2012 Configuration Manager**.

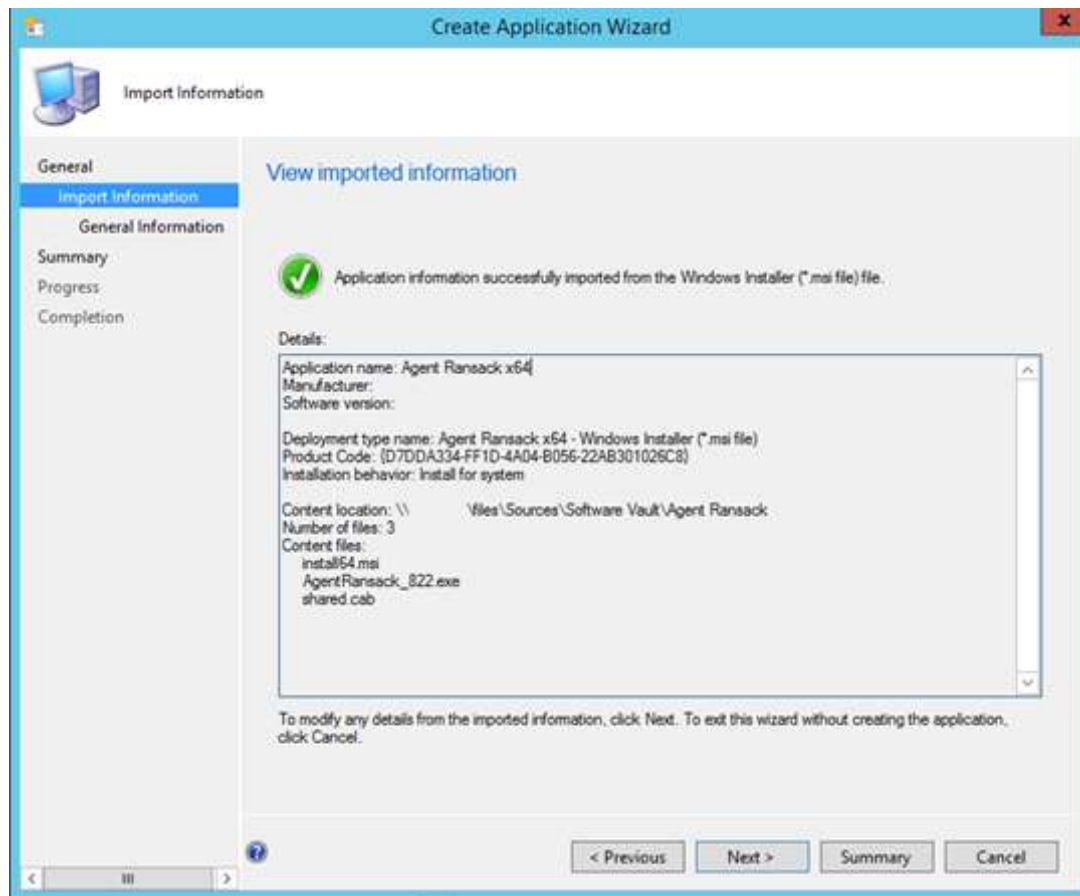
In the lower left corner, select **Software Library**, expand **Application Management**, expand **Applications**, right-click **Client Apps**, and select **Create Application**.

Paste the path to the install folder on the Software Vault share into the **Location:** field on the *Specify settings for this application* screen and click the **Browse:** button. In the **Open** file wizard, select the .MSI file and click **Open**.



Click **Next** in the wizard, and if a message popup opens indicating that the publisher cannot be verified, click **Yes**.

Click **Next** on the *View imported information* screen.



On the *Specify information about this application* screen, do the following with information from the application package document:

- Specify the application licensing type in the **Administrator comments:** field and include any other relevant details.
- Copy the **Publisher** value from Notes section into the **Manufacturer:** field.
- Copy the **DisplayVersion** value from the Notes section into the **Software version:** field.
- In the **Installation program:** field, ensure that the msixexec call has a **/qn /norestart** at the end:
 - Example: `msiexec /i "install64.msi" /qn /norestart`
- Ensure that the **Install behavior:** dropdown is set to **Install for system**.

Click **Next** in the wizard.

The screenshot shows the 'Create Application Wizard' window with the 'Specify information about this application' screen active. The left sidebar contains a navigation pane with 'General Information' selected. The main area contains the following fields and options:

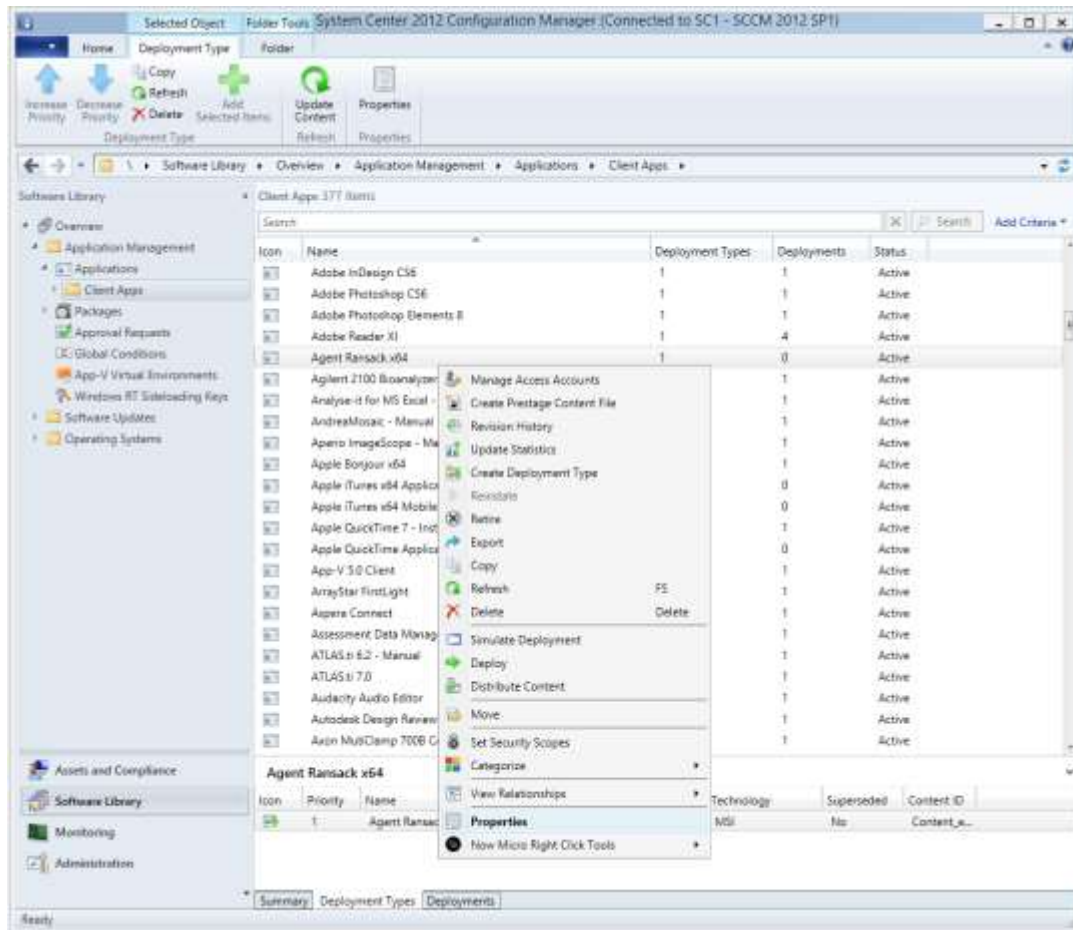
- Name:** Agent Ransack x64n
- Administrator comments:** Free software license.
- Manufacturer:** Mythicsoft Ltd
- Software version:** 7.0.822.1
- Optional reference:** (empty field)
- Administrative categories:** (empty dropdown with 'Select...' button)
- Specify the installation program for this application and the required installation rights.**
 - Installation program:** msiexec /i "install64.msi" /qn /norestart (with 'Browse...' button)
 - ☐ Run installation program as 32-bit process on 64-bit clients.
 - Install behavior:** Install for system (dropdown menu)

At the bottom, there are buttons for '< Previous', 'Next >', 'Summary', and 'Cancel'.

Click **Next** at the *Confirm the settings for this application* screen.

Click **Close** to exit the wizard.

In the SCCM **Client Apps** list, find the newly created package, right-click the name and select **Properties**.



On the **General Information** tab:

- Enable the *Allow this application to be installed from the Install Application task sequence action without being deployed* checkbox.
- Change the **Owners:** field to **Administrator**
- Change the **Support contacts:** field to **Administrator**

The screenshot shows the 'Agent Ransack x64 Properties' dialog box with the 'General Information' tab selected. The 'Security' tab is also visible. The 'Name' field is 'Agent Ransack x64'. The 'Administrator comments' field contains 'Free software license.'. The 'Manufacturer' field is 'Mythicsoft Ltd' and the 'Software version' field is '7.0.822.1'. The 'Optional reference' field is empty. The 'Administrative categories' field is empty with a 'Select...' button. The 'Date published' checkbox is unchecked, and the date is '8/20/2014'. The checkbox 'Allow this application to be installed from the Install Application task sequence action without being deployed' is checked. Below this, the 'Specify the administrative users who are responsible for this application.' section shows 'Owners:' as 'Administrator' and 'Support contacts:' as 'Administrator', both with 'Browse...' buttons. At the bottom, a summary table shows: Created date: 8/20/2014 10:50 AM, Created by: DOMAIN\userName, Modified date: 8/20/2014 10:52 AM, Modified by: DOMAIN\userName, Revision: 2, Status: Active, Superseded: No. The 'OK', 'Cancel', and 'Apply' buttons are at the bottom right.

Specify the administrative users who are responsible for this application.	
Owners:	Administrator
Support contacts:	Administrator

Created date:	8/20/2014 10:50 AM	Revision:	2
Created by:	DOMAIN\userName	Status:	Active
Modified date:	8/20/2014 10:52 AM	Superseded:	No
Modified by:	DOMAIN\userName		

On the **Application Catalog** tab:

- Ensure that the **Localized description:** field indicates the application licensing type and any important installation notes.

The screenshot shows the 'Agent Ransack x64 Properties' dialog box with the 'Application Catalog' tab selected. The dialog has a title bar with a close button (X) and a help button (?). The tabs are: Security, General Information, Application Catalog (selected), References, Distribution Settings, Deployment Types, Content Locations, and Supersede. The main area contains instructions: 'Specify information about how you want to display this application to users when they browse the Application Catalog. To provide information in a specific language, select the language before you enter a description.' Below this, there is a 'Selected language:' dropdown menu set to 'English (United States) default' and an 'Add/Remove...' button. The 'Localized application name:' field contains 'Agent Ransack x64'. The 'User categories:' field is empty, with an 'Edit...' button. The 'User documentation:' field is empty, with a 'Browse...' button. The 'Link text:' field is empty. The 'Localized description:' field contains 'Free software license.' and has a vertical scrollbar. The 'Keywords:' field is empty. The 'Icon:' field shows a small icon and a 'Browse...' button. At the bottom, there are 'OK', 'Cancel', and 'Apply' buttons.

Agent Ransack x64 Properties

Security | General Information | **Application Catalog** | References | Distribution Settings | Deployment Types | Content Locations | Supersede

Specify information about how you want to display this application to users when they browse the Application Catalog. To provide information in a specific language, select the language before you enter a description.

Selected language: English (United States) default Add/Remove...

Localized application name: Agent Ransack x64

User categories: Edit...

User documentation: Browse...

Link text:

Localized description: Free software license.

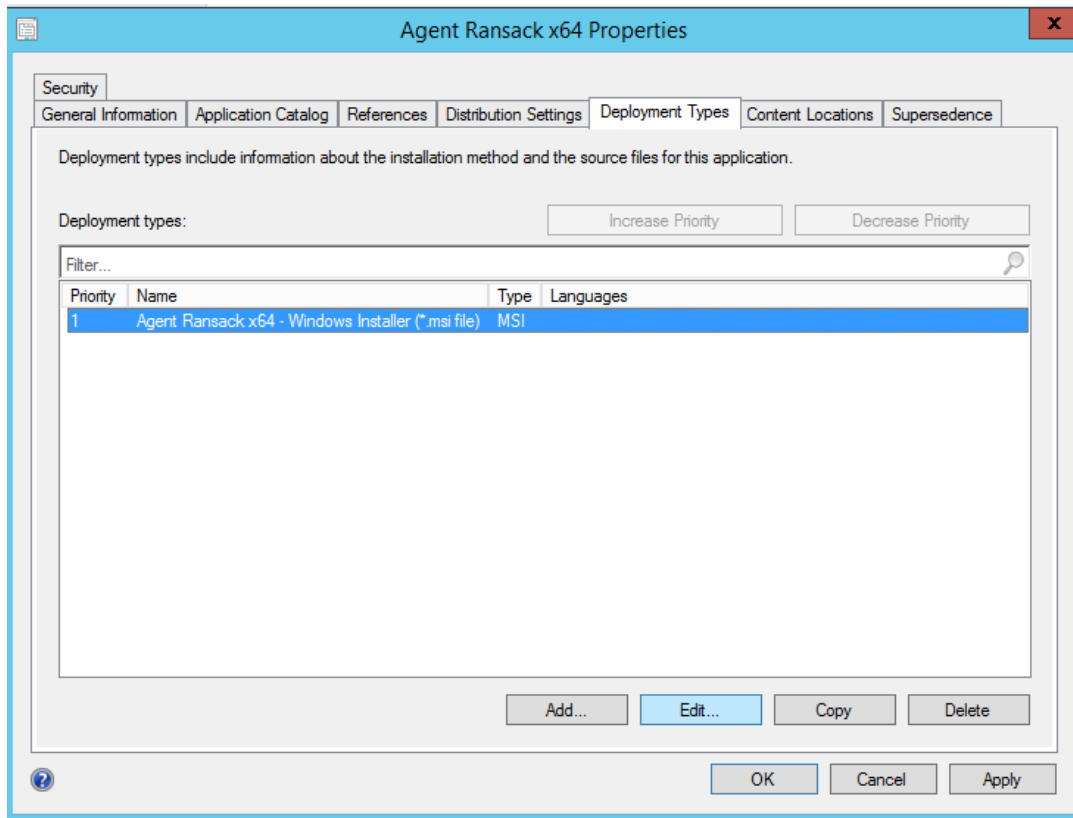
Keywords:

Icon: Browse...

OK Cancel Apply

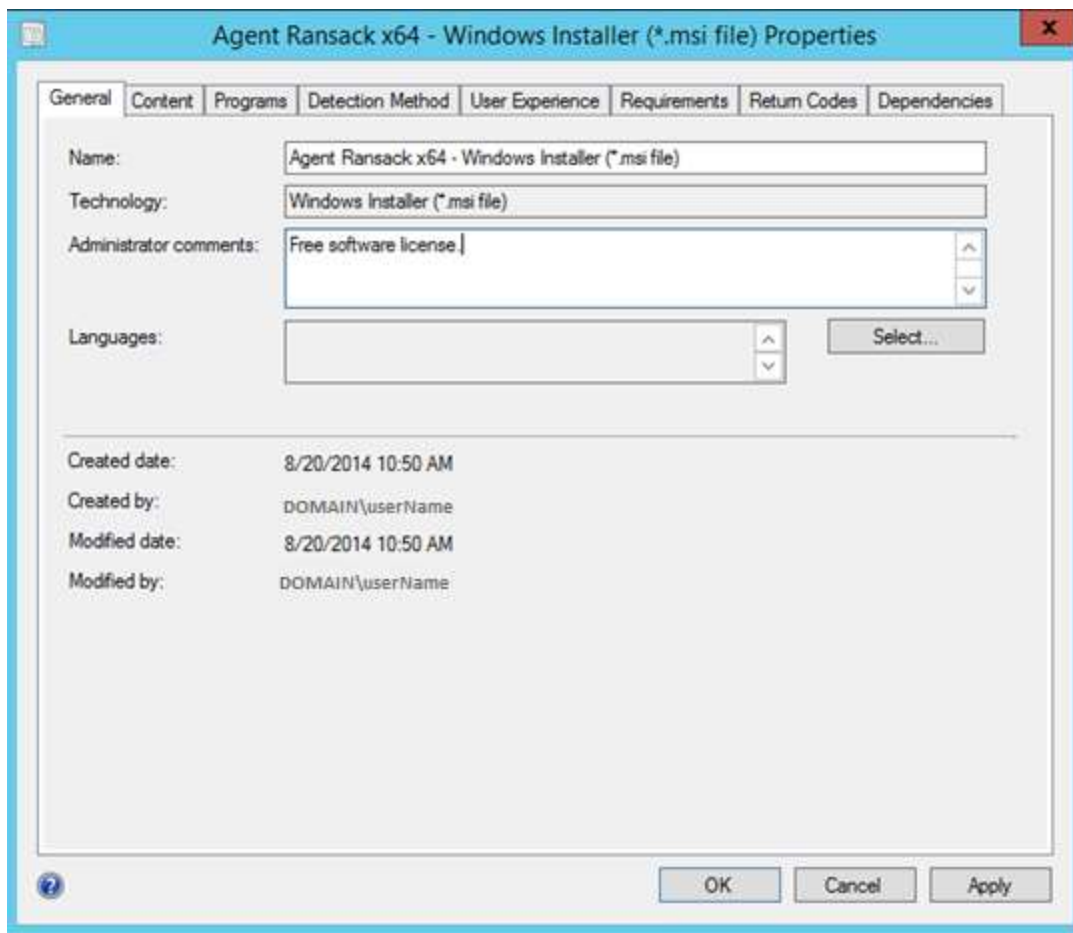
On the **Deployment Types** tab:

- Select the name of the MSI deployment type and click the **Edit...** button.



On *Windows installer (*.msi file) Properties* page, do the following:

- On the **General** tab, add the application licensing type and any important installation notes to the **Administrator comments:** field.



- On the **Content** page, uncheck the *Allow clients to share content with other clients on the same subnet* checkbox.

The screenshot shows the 'Content' tab of the 'Agent Ransack x64 - Windows Installer (*.msi file) Properties' dialog box. The 'Content location' is set to '\\files\Sources\Software Vault\Agent Ransack'. The checkbox 'Persist content in the client cache' is unchecked. The checkbox 'Allow clients to share content with other clients on the same subnet' is also unchecked. Below this checkbox, a note states: 'This option allows clients that use Windows BranchCache to download content from on-premises distribution points. Content downloads from cloud-based distribution points can always be shared by clients that use Windows BranchCache.' The checkbox 'Allow clients to use a fallback source location for content' is unchecked. Below this checkbox, a note states: 'When this content is not available on any preferred distribution points, you can allow the client to use a fallback source location for content.' The 'Deployment options' dropdown menu is set to 'Do not download content'. The 'OK', 'Cancel', and 'Apply' buttons are at the bottom right.

Agent Ransack x64 - Windows Installer (*.msi file) Properties

General Content Programs Detection Method User Experience Requirements Return Codes Dependencies

Specify the content location and delivery options for this deployment type.

Content location: \\files\Sources\Software Vault\Agent Ransack Browse...

☐ Persist content in the client cache

☐ Allow clients to share content with other clients on the same subnet

This option allows clients that use Windows BranchCache to download content from on-premises distribution points. Content downloads from cloud-based distribution points can always be shared by clients that use Windows BranchCache.

☐ Allow clients to use a fallback source location for content

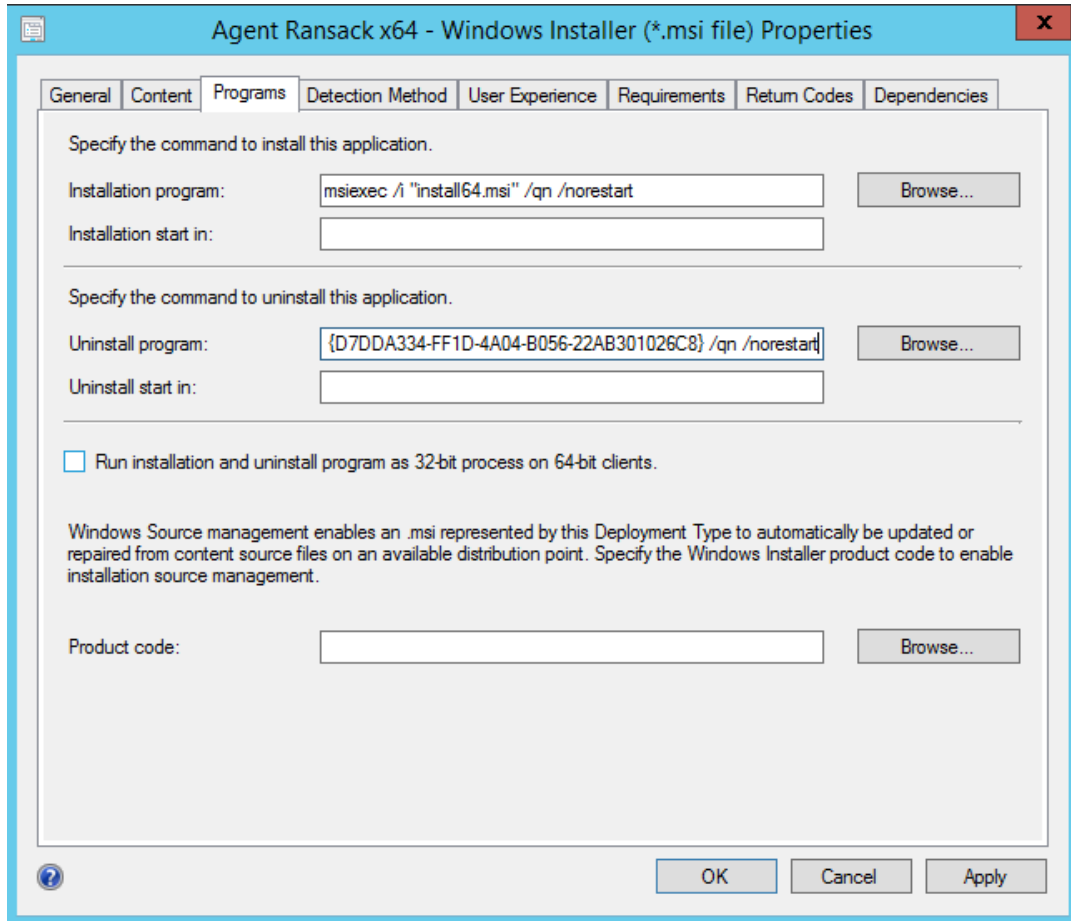
When this content is not available on any preferred distribution points, you can allow the client to use a fallback source location for content.

Select the deployment option to use when a client is within a slow or unreliable network boundary, or when the client uses a fallback source location for content.

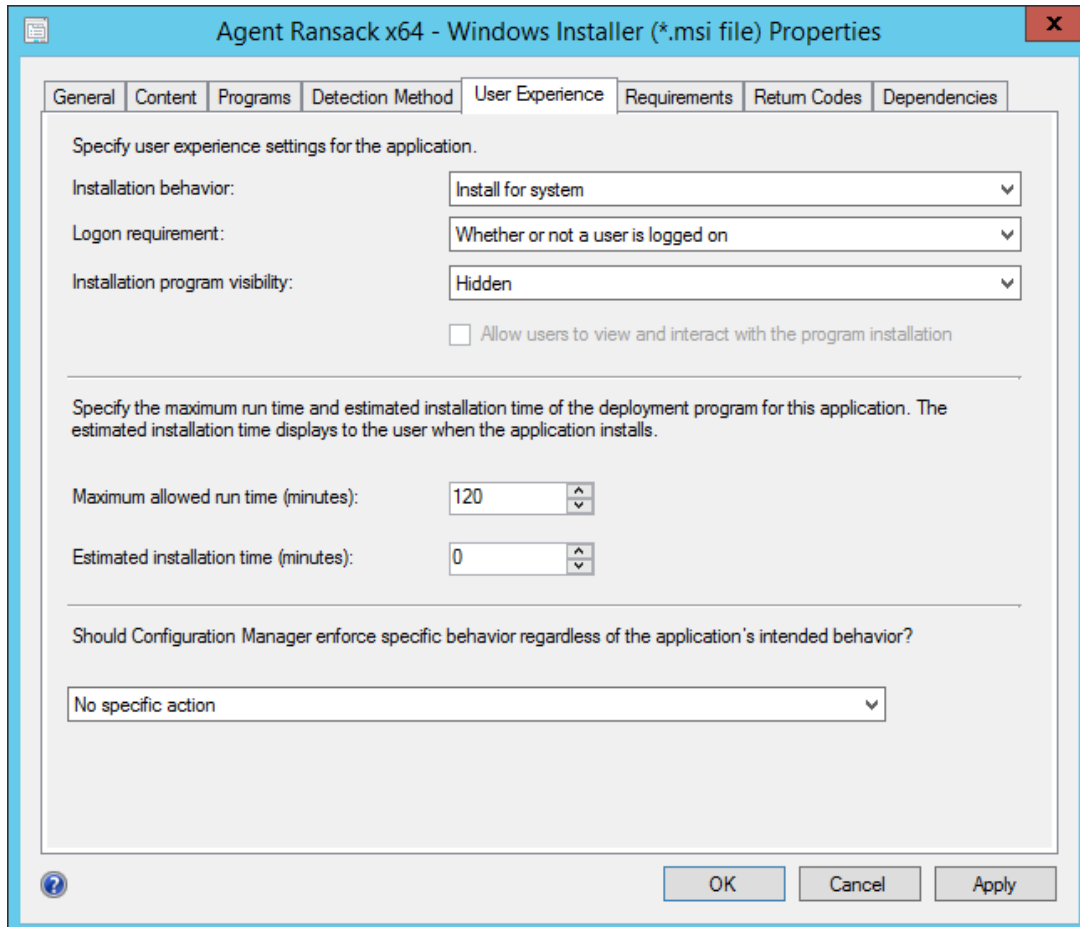
Deployment options: Do not download content

OK Cancel Apply

- On the **Programs** tab, add **/qn /norestart** to the end of the string in the **Uninstall program:** field.



- On the **User Experience** tab, do the following:
 - Installation behavior:** should be set to **Install for system**
 - Logon requirement:** should be set to **Whether or not a user is logged on**
 - Installation program visibility:** should be set to **Hidden**
 - The application behavior dropdown should be set to **No specific action**



Click **OK** to close the *Windows Installer (*.msi file) Properties* page.

Click **OK** to close the application properties page.

Find the application package in the SCCM **Client Apps** list and move on to the **4. How to Distribute Content** section.

Note: If an application takes a long time to install, set **Visibility** to **Normal** and use the /qb switch with the msixexec command in the **Install program** field from the previous step to make the progress bar visible so technicians can see that the installation is running without having to open the Task Manager to search for the process.

3. How to Package Script Installers

Script installer packages are used for installers that use an executable file (.EXE) or require a PowerShell program. All PowerShell program calls need to be in the following format to execute properly on the client system.

```
Powershell.exe -executionpolicy Bypass -file "FileName.ps1"
```

To create the Script Installer:

Open a Remote Desktop session to the server running SCCM and log in using a Domain Admin user account.

Start the **System Center 2012 Configuration Manager**.

In the lower left corner, select **Software Library**, expand **Application Management**, expand **Applications**, right-click **Client Apps** and select **Create Application**.

Select the **Manually specify the application information** button on the *Specify settings for this application* page. Click **Next** in the wizard.

On the *Specify information about this application* screen, do the following with information from the application package document:

- Specify the application's name in the **Name:** field. If the package requires user interaction to install, add - **Manual** after the name.
- Specify the application licensing type and any important installation notes in the **Administrator comments:** field.
- Copy the **Publisher** value from Notes section into the **Manufacturer:** field.
- Copy the **DisplayVersion** value from the Notes section into the **Software version:** field.
- Enable the **Allow this application to be installed from the Install Application task sequence without being deployed** checkbox.
- Change the name in the **Owners:** field to **Administrator**
- Change the name in the **Support contacts:** field to **Administrator**

Click **Next** in the wizard.

The screenshot shows the 'Create Application Wizard' dialog box with the 'General Information' tab selected. The left sidebar contains a tree view with 'General' expanded, showing 'General Information', 'Application Catalog', 'Deployment Types', 'Summary', 'Progress', and 'Completion'. The main area is titled 'Specify information about this application' and contains the following fields and controls:

- Name:** EndoScan V 4 SP1
- Administrator comments:** Single license package. Registration Code required to activate software. Choose No at the system restart popup message.
- Manufacturer:** Charles River
- Software version:** 4.1
- Optional reference:** (empty field)
- Administrative categories:** (dropdown menu with up/down arrows and a 'Select...' button)
- Date published:** 9/ 5/2014 (checkbox and date field)
- ☒ **Allow this application to be installed from the Install Application task sequence action without being deployed**
- Specify the administrative users who are responsible for this application.**
- Owners:** Administrator (text field with a 'Browse...' button)
- Support contacts:** Administrator (text field with a 'Browse...' button)

At the bottom, there is a help icon and four buttons: '< Previous', 'Next >', 'Summary', and 'Cancel'.

On the *Specify the Configuration Manager Application Catalog entry* screen, do the following:

- For packages requiring user interaction to install, ensure that the **Localized application name:** field includes a - **Manual** after the application name.
- Specify the application licensing type and any important installation notes in the **Localized description:** field.

Click **Next** in the wizard.

The screenshot shows the 'Create Application Wizard' window with the 'Specify the Configuration Manager Application Catalog entry' screen selected. The left sidebar contains a tree view with 'General' expanded, showing 'General Information', 'Application Catalog' (selected), 'Deployment Types', 'Summary', 'Progress', and 'Completion'. The main area has a title bar 'Create Application Wizard' and a subtitle 'Specify the Configuration Manager Application Catalog entry'. Below the subtitle is a description: 'Specify information about how you want to display this application to users when they browse the Application Catalog. To provide information in a specific language, select the language before you enter a description.' The 'Selected language:' dropdown is set to 'English (United States) default' with an 'Add/Remove...' button. The 'Localized application name:' field contains 'EndoScan V 4 SP1'. The 'User categories:' field is empty with an 'Edit...' button. The 'User documentation:' field is empty with a 'Browse...' button. The 'Link text:' field is empty. The 'Localized description:' field contains the text 'Single license package. Registration Code required to activate software. Choose No at the system restart popup message.' The 'Keywords:' field is empty. The 'Icon:' field shows a default application icon with a 'Browse...' button. At the bottom are buttons for '< Previous', 'Next >', 'Summary', and 'Cancel'.

Application Catalog

General
General Information
Application Catalog
Deployment Types
Summary
Progress
Completion

Specify the Configuration Manager Application Catalog entry

Specify information about how you want to display this application to users when they browse the Application Catalog. To provide information in a specific language, select the language before you enter a description.

Selected language: English (United States) default Add/Remove...

Localized application name: EndoScan V 4 SP1

User categories: Edit...

User documentation: Browse...

Link text:

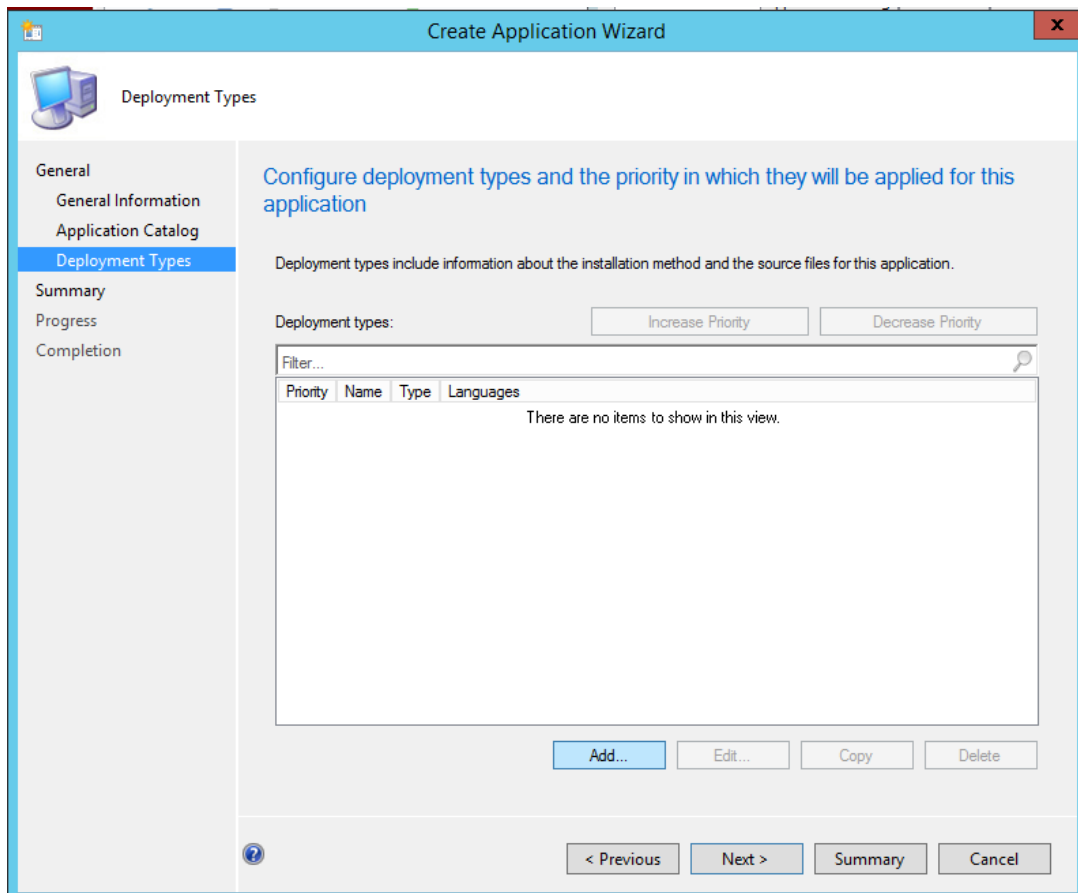
Localized description: Single license package. Registration Code required to activate software. Choose No at the system restart popup message.

Keywords:

Icon: Browse...

< Previous Next > Summary Cancel

On the *Configure deployment types and the priority in which they will be applied for this application* screen, click the **Add...** button to start the **Create Deployment Type Wizard**.



On the *Specify settings for this deployment type* screen in the **Create Deployment Type Wizard**, select **Script Installer** from the **Type:** drop-down list. The **Manually specify the deployment type information** button will automatically be selected. Click **Next** in the wizard.

Create Deployment Type Wizard

General

General

Import Information
General Information
Requirements
Dependencies
Summary
Progress
Completion

Specify settings for this deployment type

Deployment types include information about the installation method and source files for this application.

Type: **Script Installer**

☒ Automatically specify deployment type information

Location:

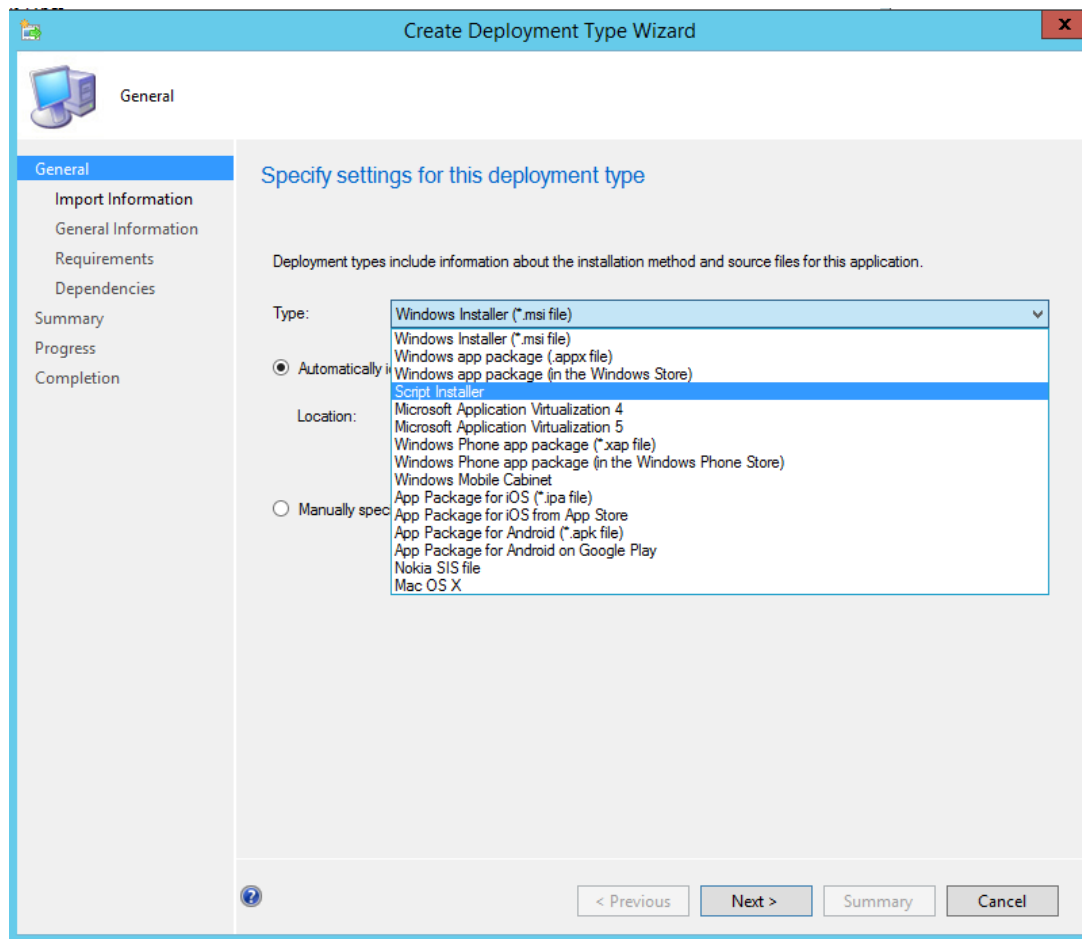
☐ Manually specify deployment type information

< Previous **Next >** Summary Cancel

On the *Specify general information for this deployment type* screen, do the following:

- Provide a meaningful name for the setup package in the **Name:** field.
- Specify the application licensing type and any important installation notes in the **Administrator comments:** field.

Click **Next** in the wizard.



(Incorrect screenshot – duplicate of last screenshot)

On the *Specify information about the content to be delivered to target devices* screen, do the following:

- Click the **Browse...** button for **Content location:** and browse to the location of the application's setup files on the **Software Vault** share.
- Disable the **Allow clients to share content with other clients on the same subnet** checkbox.
- Click the **Browse...** button for **Installation program:** and select the .EXE file or PowerShell program that starts the application installer.
- Click the **Browse...** button for **Uninstall program:** and select the PowerShell program that uninstalls the application and its components.

Note: Any PowerShell programs called as the **Install program** or **Uninstall program** need to be in the following format to execute properly on the client system.

```
Powershell.exe -executionpolicy Bypass -file "FileName.ps1"
```

The screenshot shows the 'Create Deployment Type Wizard' window, specifically the 'Content' tab. The window has a blue title bar and a sidebar on the left with the following options: General, General Information, Content (selected), Detection Method, User Experience, Requirements, Dependencies, Summary, Progress, and Completion. The main area is titled 'Specify information about the content to be delivered to target devices'. It contains the following fields and controls:

- Content location:** A text box containing 'ources\Software Vault\EndoScan\ESV 4 SP1 (D)' and a 'Browse...' button.
- ☐ **Persist content in the client cache**
- ☐ **Allow clients to share content with other clients on the same subnet**
This option allows clients that use Windows BranchCache to download content from on-premises distribution points. Content downloads from cloud-based distribution points can always be shared by clients that use Windows BranchCache.
- Specify the command used to install this content.**
- Installation program:** A text box containing '"EndoScanVSetup.exe" /SILENT' and a 'Browse...' button.
- Installation start in:** An empty text box.
- Configuration Manager can remove installations of this content if an uninstall program is specified below.**
- Uninstall program:** A text box containing '.tionpolicy Bypass -file "EndoScanV-Uninstall.ps1"' and a 'Browse...' button.
- Uninstall start in:** An empty text box.
- ☐ **Run installation and uninstall program as 32-bit process on 64-bit clients.**

At the bottom, there are four buttons: '< Previous', 'Next >' (highlighted), 'Summary', and 'Cancel'.

On the *Specify how this deployment type is detected* screen, click the **Add Clause...** button to open the **Detection Rule** page.

The screenshot shows the 'Create Deployment Type Wizard' window, specifically the 'Detection Method' step. The window has a blue title bar and a sidebar on the left with the following options: General, General Information, Content, Detection Method (selected), User Experience, Requirements, Dependencies, Summary, Progress, and Completion. The main area is titled 'Specify how this deployment type is detected' and contains the following text: 'Specify how Configuration Manager determines whether this deployment type is already present on a device. This detection occurs before the content is installed or when software inventory data is collected.'

There are two radio buttons for selection:

- ☒ Configure rules to detect the presence of this deployment type:
- ☐ Use a custom script to detect the presence of this deployment type:

Under the first option, there is a large empty box for rules. Above this box is a header bar with 'Connector (Clause)'. To the right of the box are three buttons: 'Add Clause...' (highlighted in blue), 'Edit Clause...', and 'Delete Clause'. Below these buttons are 'Group' and 'Ungroup' buttons. A red exclamation mark icon is visible to the right of the rule box.

Under the second option, there are two text boxes: 'Script type:' and 'Script length:'. To the right of these boxes is an 'Edit...' button.

At the bottom of the window, there are four buttons: '< Previous', 'Next >', 'Summary', and 'Cancel'.

On the *Detection Rule* page, do the following:

- From the package details obtained at the beginning, determine the type of **Setting Type** to be used for detecting this application.
 - **File System** -- used to detect the existence of a file or folder on the client system.
 - **Registry** -- looks for the existence of a registry key, or some registry property value for that key.
 - **Windows Installer** -- checks to see if an application with the specified GUID exists in the registry.
- For the purposes of this example, one would do the following:
 - Select **Registry** from the **Setting Type**: drop-down list.
 - Select **HKEY_LOCAL_MACHINE** from the **Hive**: drop-down list.
 - Paste the registry key location into the **Key**: text box.

SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall\EndoScan-V_is1

- Click **OK** to exit the *Detection Rule* page.

Detection Rule

Create a rule that indicates the presence of this application.

Setting Type: Registry

Specify the registry key or value to detect this application.

Hive: HKEY_LOCAL_MACHINE

Key: SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall\EndoScan-V_is1

Value:

☐ Use (Default) registry key value for detection

☐ This registry key is associated with a 32-bit application on 64-bit systems

Data Type:

☒ This registry setting must exist on the target system to indicate presence of this application

☐ This registry setting must satisfy the following rule to indicate the presence of this application

Operator:

Value:

OK Cancel

On the *Specify how this deployment type is detected* screen, click **Next** in the wizard.

The screenshot shows the 'Create Deployment Type Wizard' window, specifically the 'Detection Method' step. The window has a blue title bar and a sidebar on the left with the following options: General, General Information, Content, Detection Method (selected), User Experience, Requirements, Dependencies, Summary, Progress, and Completion. The main area is titled 'Specify how this deployment type is detected' and contains the following text: 'Specify how Configuration Manager determines whether this deployment type is already present on a device. This detection occurs before the content is installed or when software inventory data is collected.'

There are two radio buttons for configuration:

- ☒ Configure rules to detect the presence of this deployment type:
- ☐ Use a custom script to detect the presence of this deployment type:

The first option is selected and leads to a table with two columns: 'Connector' and 'Clause'. The table has one row with a right-pointing arrow in the 'Connector' column and the text 'LocalMachine\SOFTWARE\Wow6432Node\Micr...' in the 'Clause' column. To the right of the table are three buttons: 'Add Clause...', 'Edit Clause...', and 'Delete Clause...'. Below the table are two more buttons: 'Group' and 'Ungroup'.

The second option leads to two text boxes: 'Script type:' and 'Script length:', with an 'Edit...' button to the right.

At the bottom of the window are four buttons: '< Previous', 'Next >' (highlighted in blue), 'Summary', and 'Cancel'.

Connector	Clause
▶	LocalMachine\SOFTWARE\Wow6432Node\Micr...

On the *Specify user experience settings for the application* screen, select the user experience behavior from the drop-down lists for **Installation behavior**, **Logon requirement**, and **Installation program visibility**.

- For applications that *DO NOT REQUIRE* user interaction to install, set the following options:
 - Behavior: **Install for system**
 - Logon: **Whether or not a user is logged on**
 - Visibility: **Hidden**
- For applications that *REQUIRE* user interaction to install, set the following options:
 - Behavior: **Install for user**
 - Logon: **Only when a user is logged on**
 - Visibility: **Normal**

Create Deployment Type Wizard

User Experience

General

- General Information
- Content
- Detection Method
- User Experience**
- Requirements
- Dependencies
- Summary
- Progress
- Completion

Specify user experience settings for the application

Installation behavior: Install for system

Logon requirement: Whether or not a user is logged on

Installation program visibility: Hidden

☐ Allow users to view and interact with the program installation

Specify the maximum run time and estimated installation time of the deployment program for this application. The estimated installation time displays to the user when the application installs.

Maximum allowed run time (minutes): 120

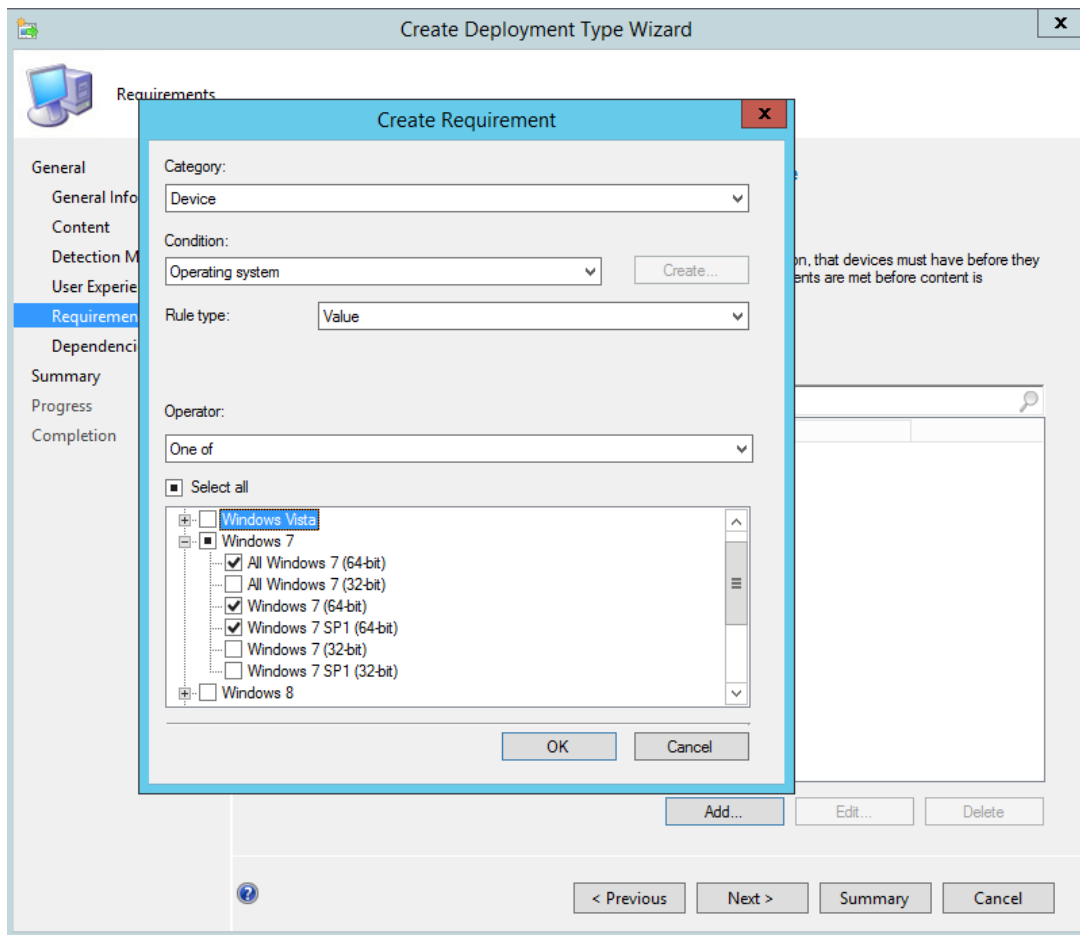
Estimated installation time (minutes): 0

< Previous Next > Summary Cancel

Note: If an application takes a long time to install, set the **Visibility** to **Normal** to make the installer visible so technicians can see the installation progress bar and know that the installation is running without having to open the Task Manager to search for the process. Stating the installation time in all Comments, Notes, and Description sections is also advisable.

On the *Specify installation requirements for this deployment type* screen, click the **Add...** button to open the **Create Requirement** page to specify any hardware features or operation system versions that devices must have before application can be installed.

For this example, only **Windows 7 (64-bit)** systems will be able to install the application.



On the *Specify installation requirements for this deployment type* screen, click **Next** in the wizard.

The screenshot shows the 'Create Deployment Type Wizard' window, specifically the 'Requirements' step. The window has a blue title bar with the text 'Create Deployment Type Wizard' and a close button. On the left is a navigation pane with the following items: 'General' (with a computer icon), 'General Information', 'Content', 'Detection Method', 'User Experience', 'Requirements' (highlighted in blue), 'Dependencies', 'Summary', 'Progress', and 'Completion'. The main area is titled 'Specify installation requirements for this deployment type'. Below this title is a paragraph: 'Specify any requirements, such as hardware features or the operating system version, that devices must have before they can install this deployment type. Configuration Manager verifies that these requirements are met before content is deployed to the device.' Below the paragraph is a section labeled 'Requirements:' containing a table with a 'Filter...' search box. The table has three columns: 'Requirement Type', 'Operator', and 'Values'. It contains one row: 'Operating system', 'One of', and '{All Windows 7 (64-bit), Windows 7 (64-bit), Win...'. At the bottom right of the table are three buttons: 'Add...', 'Edit...', and 'Delete'. At the bottom of the window are four buttons: '< Previous', 'Next >' (highlighted in blue), 'Summary', and 'Cancel'. A help icon (?) is located to the left of the 'Next >' button.

Requirements

Specify installation requirements for this deployment type

Specify any requirements, such as hardware features or the operating system version, that devices must have before they can install this deployment type. Configuration Manager verifies that these requirements are met before content is deployed to the device.

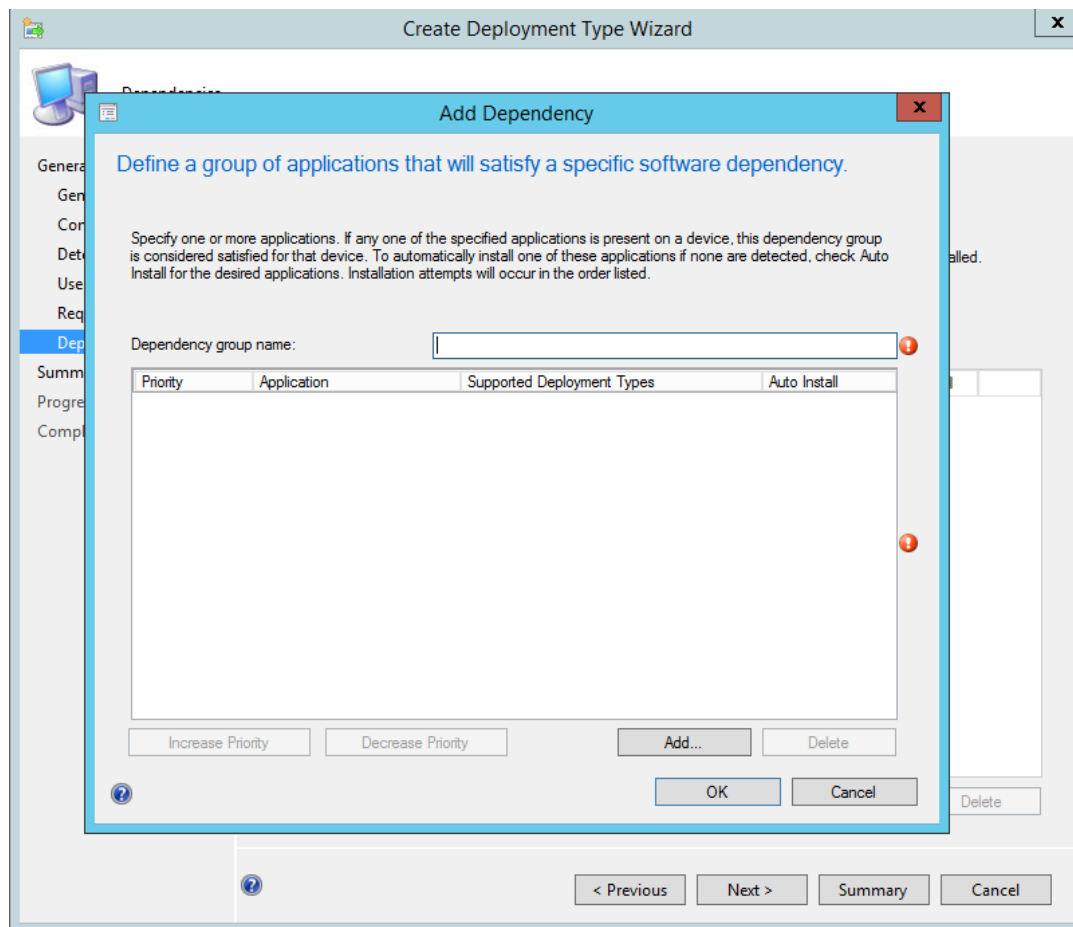
Requirements:

Requirement Type	Operator	Values
Operating system	One of	{All Windows 7 (64-bit), Windows 7 (64-bit), Win...

Buttons: Add... Edit... Delete

Navigation: < Previous Next > Summary Cancel

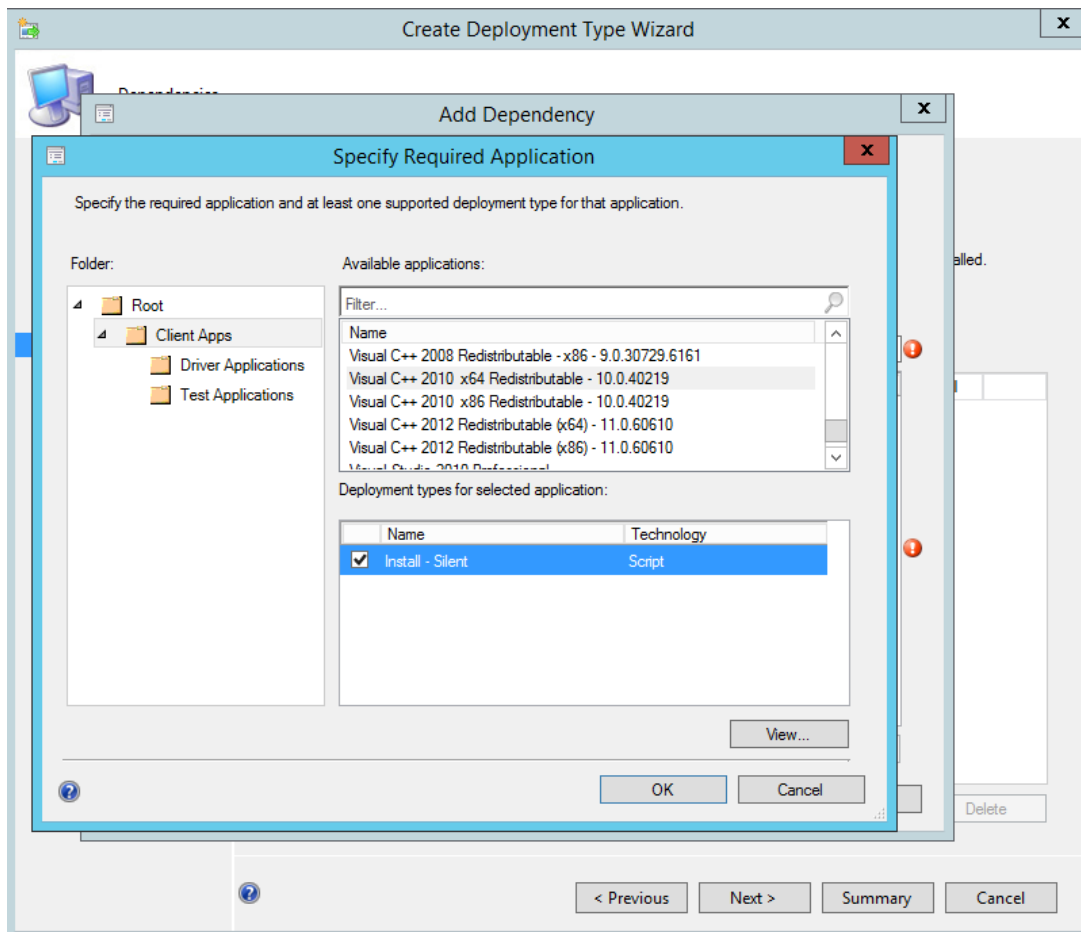
On the *Specify software dependencies for this deployment type* screen, click the **Add...** button to open the **Add Dependency** page to specify any other application packages that must be installed as prerequisites for this application package.



On the **Add Dependency** page, click the **Add...** button to choose any other application packages to install as prerequisites for installing this application package.

On the **Specify Required Application** page, do the following:

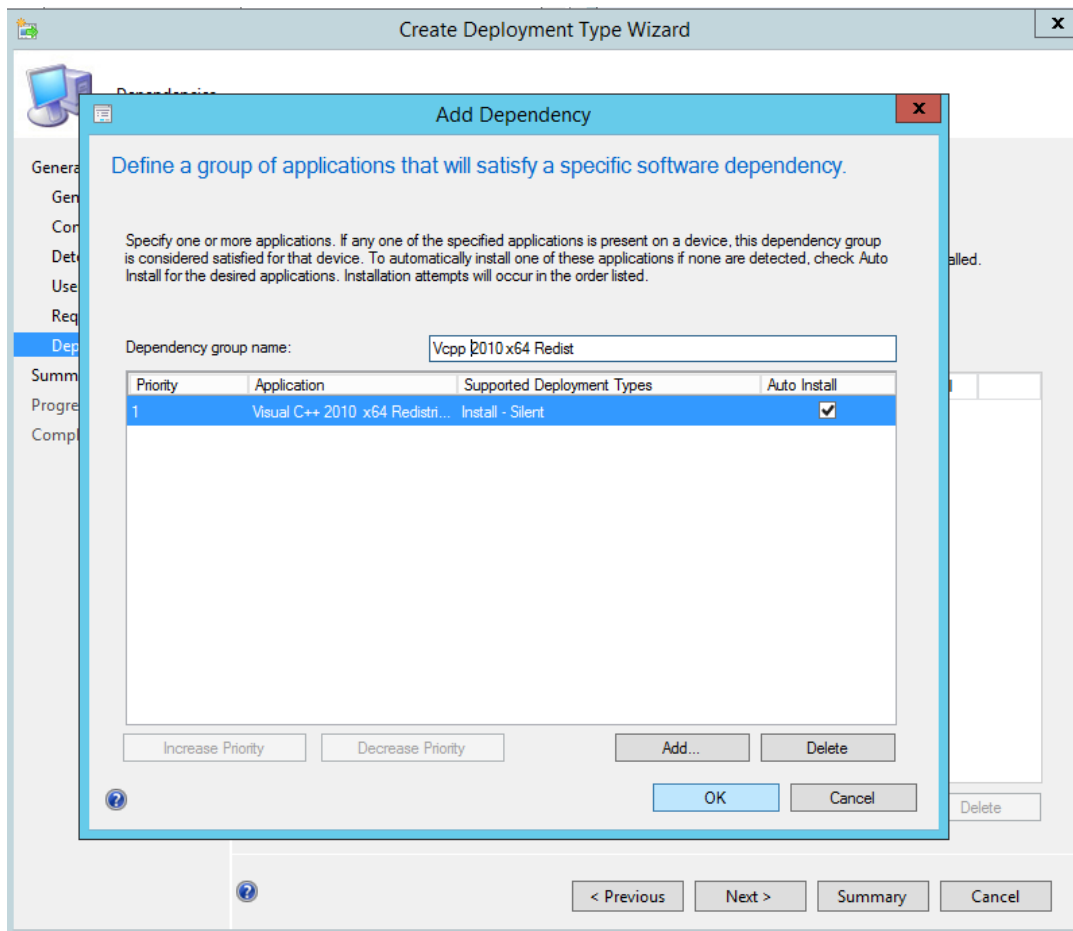
- Select **Client Apps** in the **Folder:** pane on the left.
- Find the prerequisite application package in the **Available applications:** list.
- Enable the checkbox next to the prerequisite application name in the **Deployment types for selected application:** lower, right-hand pane.
- Click **OK** to close the page.



Repeat the above steps for any other prerequisite applications packages required for installing the application being packaged.

In the **Dependency group name:** field, provide a meaningful name for the prerequisite group.

Click **OK** to close the **Add Dependency** page.



On the **Add Dependency** page, do the following:

- Click the **Add...** button to choose more application packages to install as prerequisites for installing this application package.
- Or choose **Next** in the wizard.

The screenshot shows the 'Create Deployment Type Wizard' window, specifically the 'Dependencies' page. The window has a blue title bar with the text 'Create Deployment Type Wizard' and a close button. On the left, there is a sidebar with a tree view containing the following items: 'General' (with sub-items 'General Information', 'Content', 'Detection Method', 'User Experience', 'Requirements'), 'Dependencies' (selected), 'Summary', 'Progress', and 'Completion'. The main area of the window is titled 'Specify software dependencies for this deployment type'. Below this title, there is a text box stating: 'Software dependencies are deployment types that must be installed before this deployment type can be installed.' Below the text box, there is a table titled 'Software dependencies:'. The table has five columns: 'Name', 'Application', 'Deployment Type', 'Automatically Install', and an empty column. The table contains one row with the following data: 'Vcpp 2010 x6', 'Visual C++ 2010 x64 Re', 'Install - Silent', 'Yes', and an empty cell. Below the table, there are three buttons: 'Add...', 'Edit...', and 'Delete'. At the bottom of the window, there are four buttons: '< Previous', 'Next >', 'Summary', and 'Cancel'. A help icon (?) is located at the bottom left of the main area.

Dependencies

Specify software dependencies for this deployment type

Software dependencies are deployment types that must be installed before this deployment type can be installed.

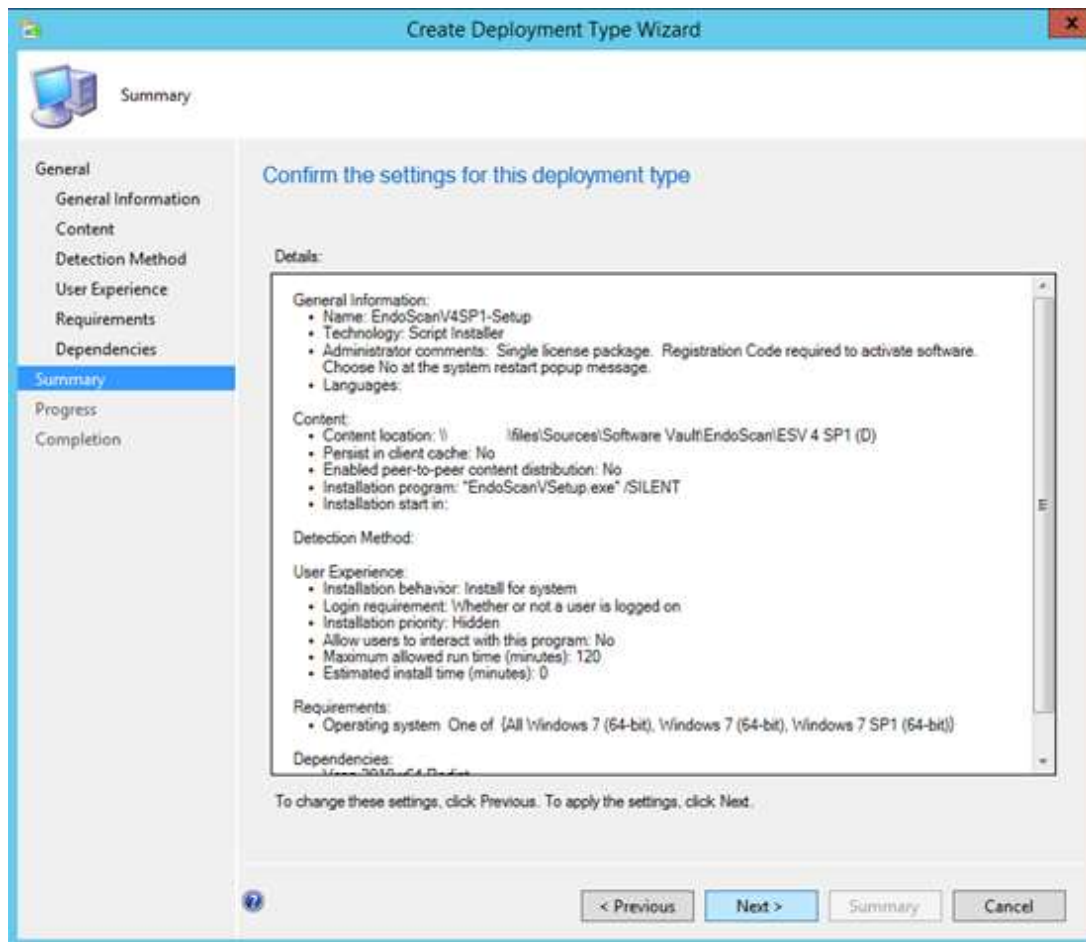
Software dependencies:

	Name	Application	Deployment Type	Automatically Install	
	Vcpp 2010 x6	Visual C++ 2010 x64 Re	Install - Silent	Yes	

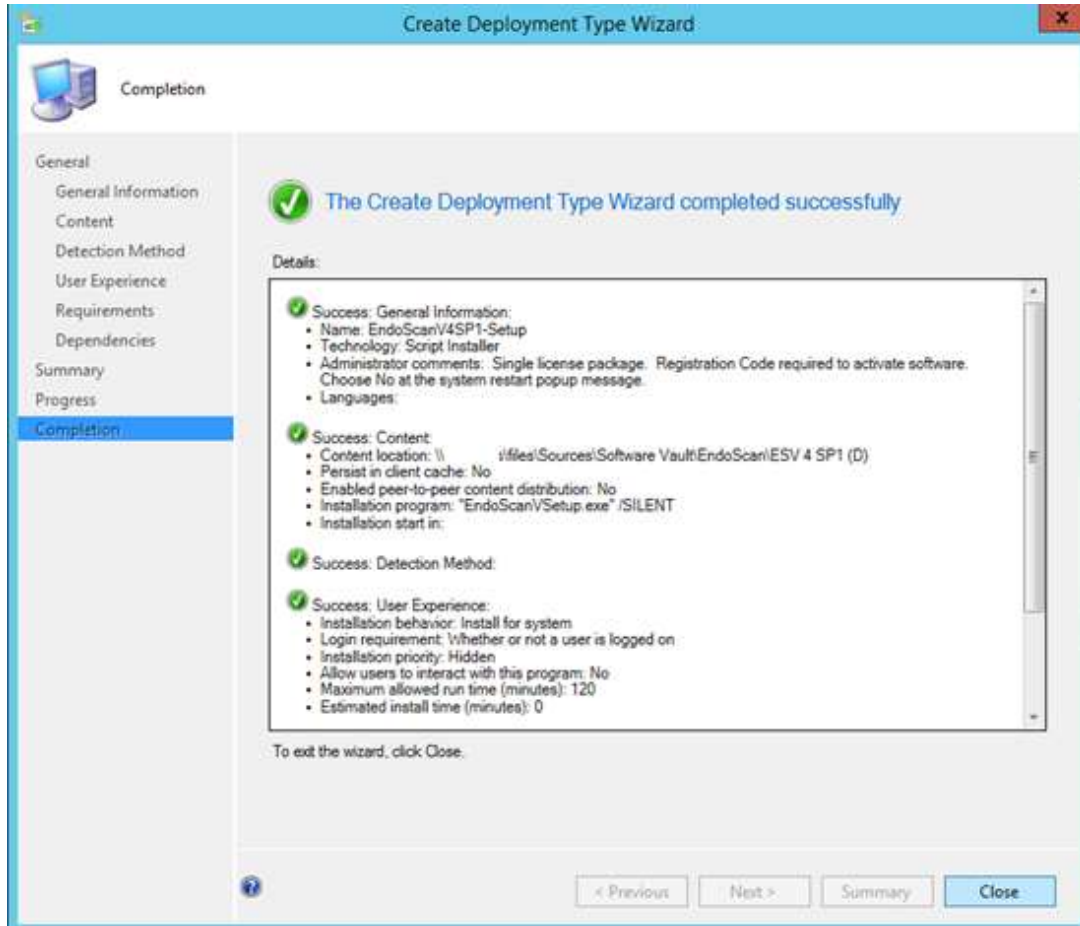
Add... Edit... Delete

< Previous Next > Summary Cancel

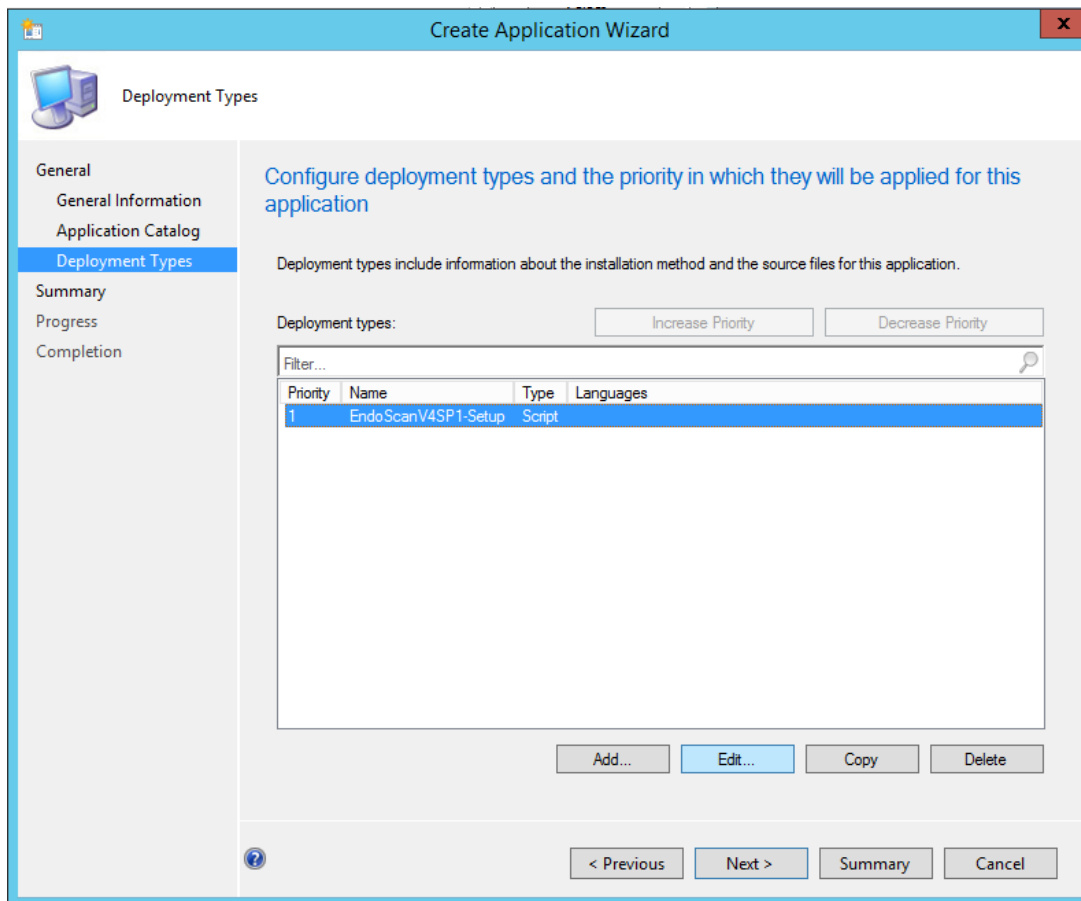
On the *Confirm the settings for this deployment type* screen, click **Next** in the wizard.



On the *Completion* screen, click **Close** to close the **Create Deployment Type Wizard**.



On the *Configure deployment types and the priority in which they will be applied for this application* screen, highlight the newly created deployment type and click the **Edit...** button.



On the *Properties* page, click the **User Experience** tab and do the following:

- Change the application behavior drop-down under **Should configuration Manager enforce specific behavior regardless of the application's intended behavior?** to **No specific action**.
- Click **OK** to exit the *Properties* page.

The screenshot shows the 'Create Application Wizard' dialog box with the 'User Experience' tab selected. The dialog is titled 'EndoScanV4SP1-Setup Properties'. On the left, a sidebar lists the wizard steps: General, General Info, Application, Deployment (highlighted), Summary, Progress, and Completion. The 'User Experience' tab contains the following settings:

- Specify user experience settings for the application.**
 - Installation behavior: Install for system
 - Logon requirement: Whether or not a user is logged on
 - Installation program visibility: Hidden
 - ☐ Allow users to view and interact with the program installation
- Specify the maximum run time and estimated installation time of the deployment program for this application. The estimated installation time displays to the user when the application installs.**
 - Maximum allowed run time (minutes): 120
 - Estimated installation time (minutes): 0
- Should Configuration Manager enforce specific behavior regardless of the application's intended behavior?**
 - Drop-down menu options: No specific action (selected), Determine behavior based on return codes, No specific action, The software install program might force a device restart, Configuration Manager client will force a mandatory device restart.

At the bottom of the dialog are buttons for '?', 'OK', 'Cancel', and 'Apply'. Below the dialog, at the bottom of the wizard window, are buttons for '< Previous', 'Next >', 'Summary', and 'Cancel'.

On the *Configure deployment types and the priority in which they will be applied for this application* screen, click **Next** in the wizard.

The screenshot shows the 'Create Application Wizard' window, specifically the 'Deployment Types' step. The window has a blue title bar with the text 'Create Application Wizard' and a close button. On the left, there is a sidebar with a tree view containing the following items: 'General' (with sub-items 'General Information' and 'Application Catalog'), 'Deployment Types' (which is selected and highlighted in blue), 'Summary', 'Progress', and 'Completion'. The main area of the window has a light blue header with the text 'Configure deployment types and the priority in which they will be applied for this application'. Below this, there is a paragraph: 'Deployment types include information about the installation method and the source files for this application.' The main area also contains a table with the following columns: 'Priority', 'Name', 'Type', and 'Languages'. The table has one row with the following data: '1', 'EndoScanV4SP1-Setup', 'Script', and an empty cell. Above the table, there is a 'Filter...' search box. To the right of the table, there are two buttons: 'Increase Priority' and 'Decrease Priority'. Below the table, there are four buttons: 'Add...', 'Edit...' (which is highlighted with a dashed border), 'Copy', and 'Delete'. At the bottom of the window, there are four buttons: '< Previous', 'Next >' (which is highlighted in blue), 'Summary', and 'Cancel'. There is also a help icon (a question mark in a circle) located to the left of the bottom buttons.

Deployment Types

General
General Information
Application Catalog
Deployment Types
Summary
Progress
Completion

Configure deployment types and the priority in which they will be applied for this application

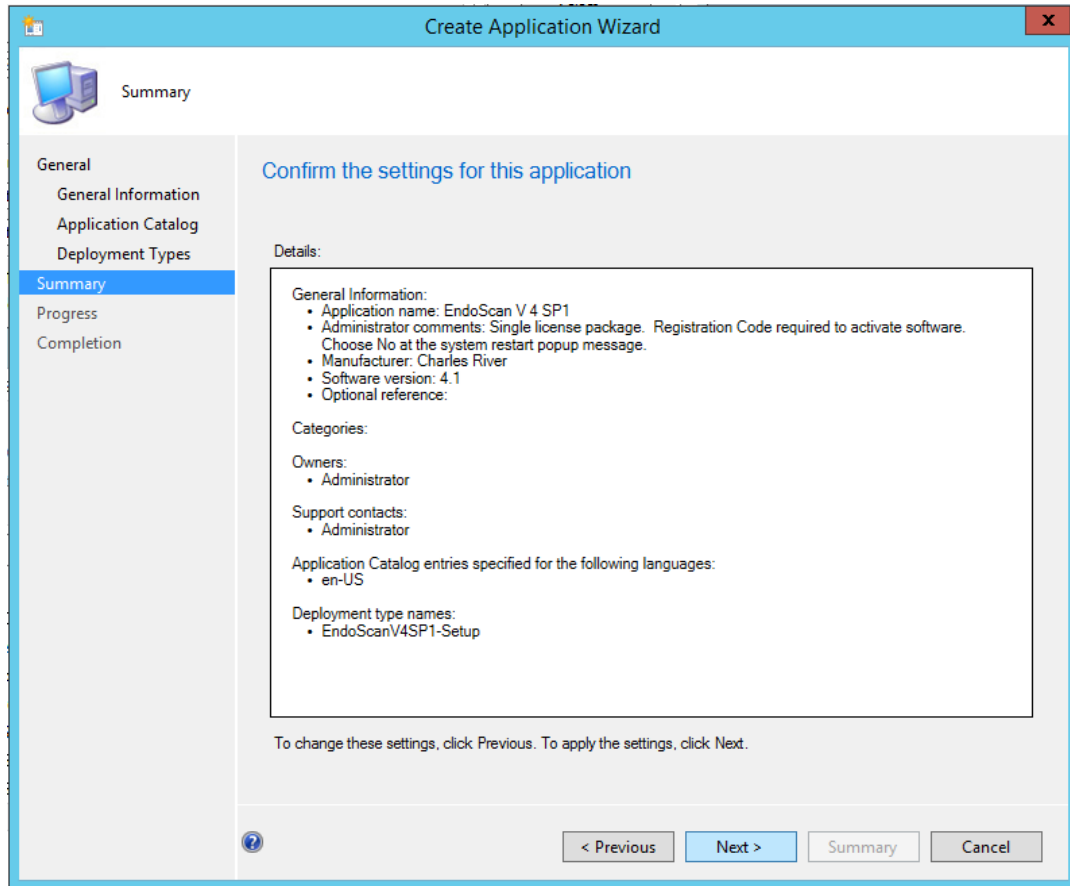
Deployment types include information about the installation method and the source files for this application.

Deployment types:

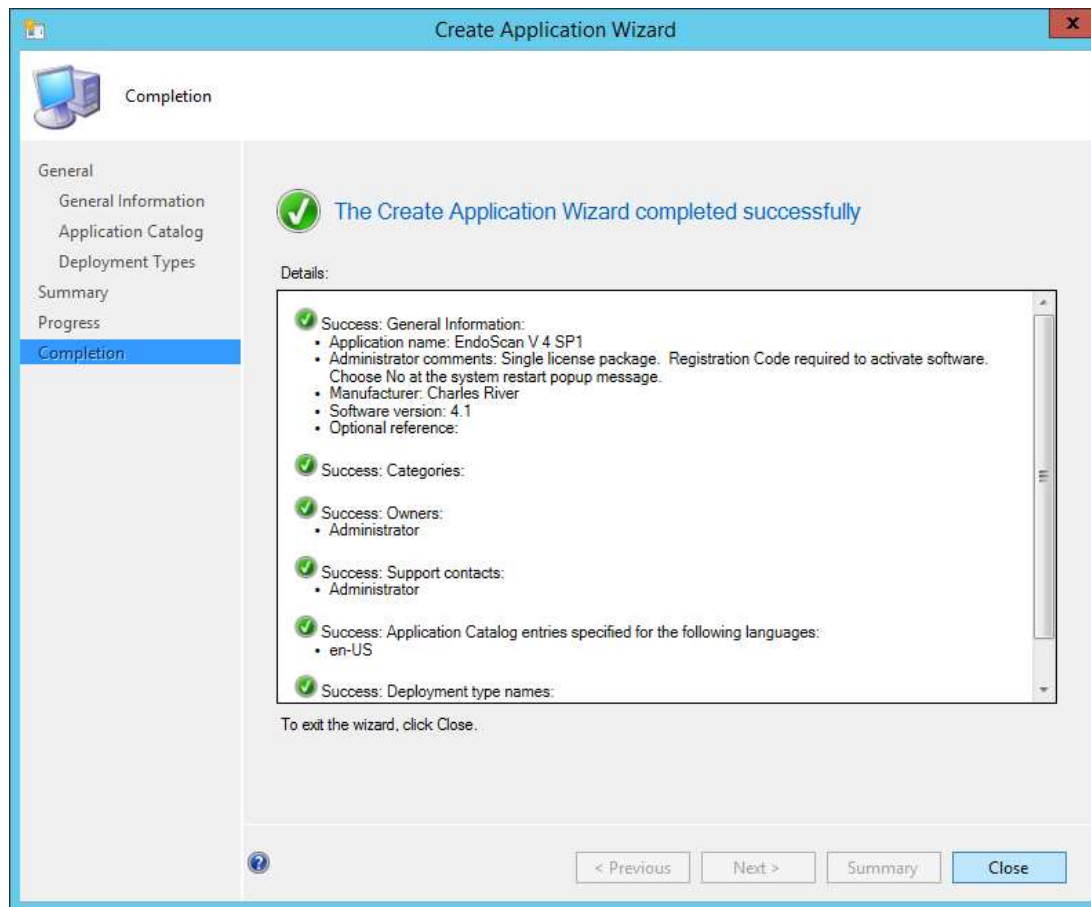
Filter...

Priority	Name	Type	Languages
1	EndoScanV4SP1-Setup	Script	

On the *Confirm the settings for this application* screen, click **Next** in the wizard.



On the *Completion* screen, click **Close** to exit the **Create Application Wizard**.

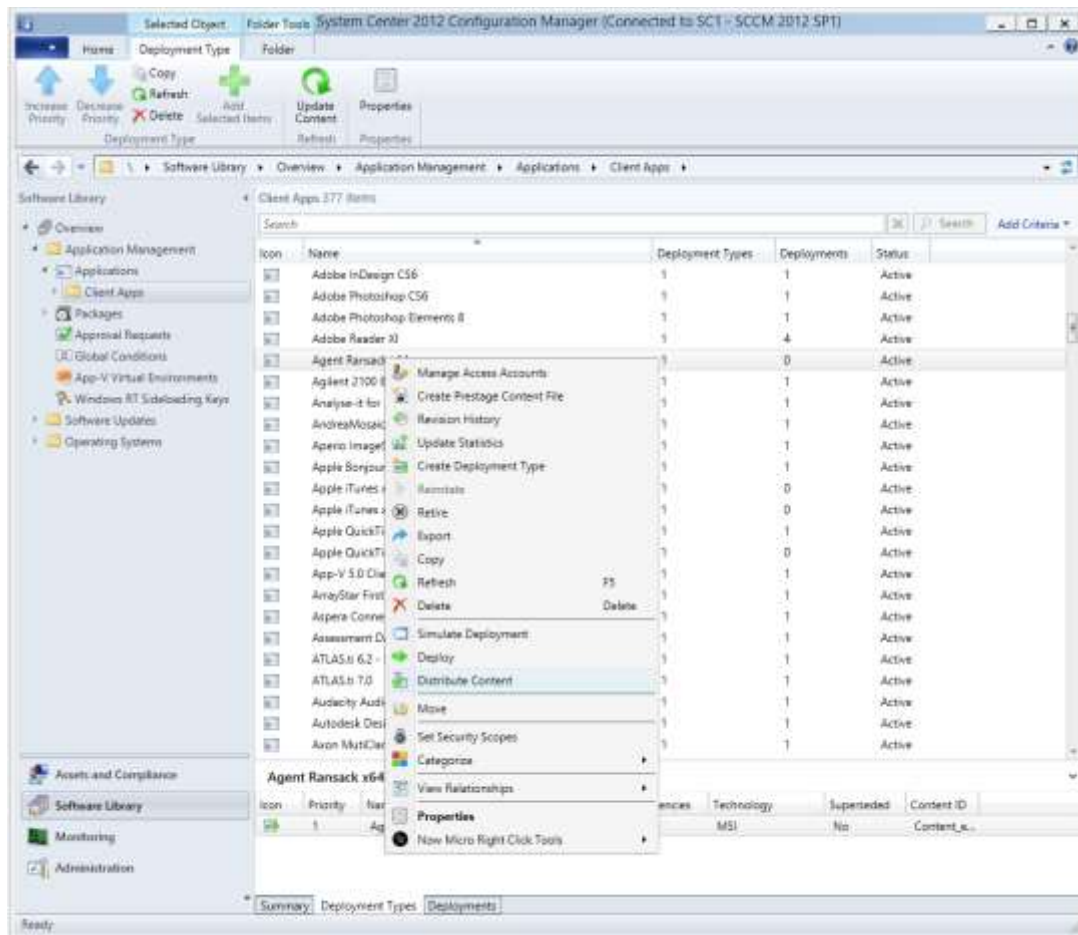


Find the application package in the SCCM **Client Apps** list and move on to the **4. How to Distribute Content** section.

4. How to Distribute Content

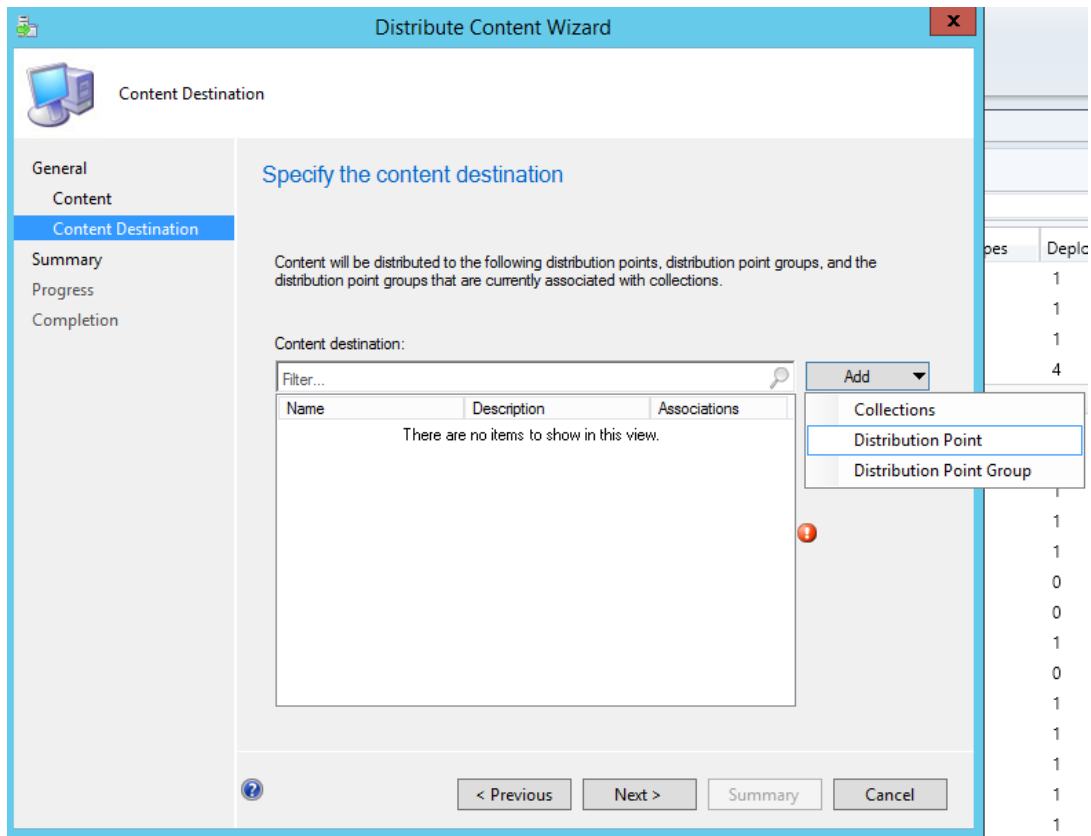
Close any open application files, like PowerShell uninstall programs, in the Software Vault application folder before attempting to distribute content to the Distribution Servers. If a file in the Software Vault application folder is open, SCCM cannot distribute the content.

Find the application package in the SCCM **Client Apps** list, right-click the package name, and select **Distribute Content**.

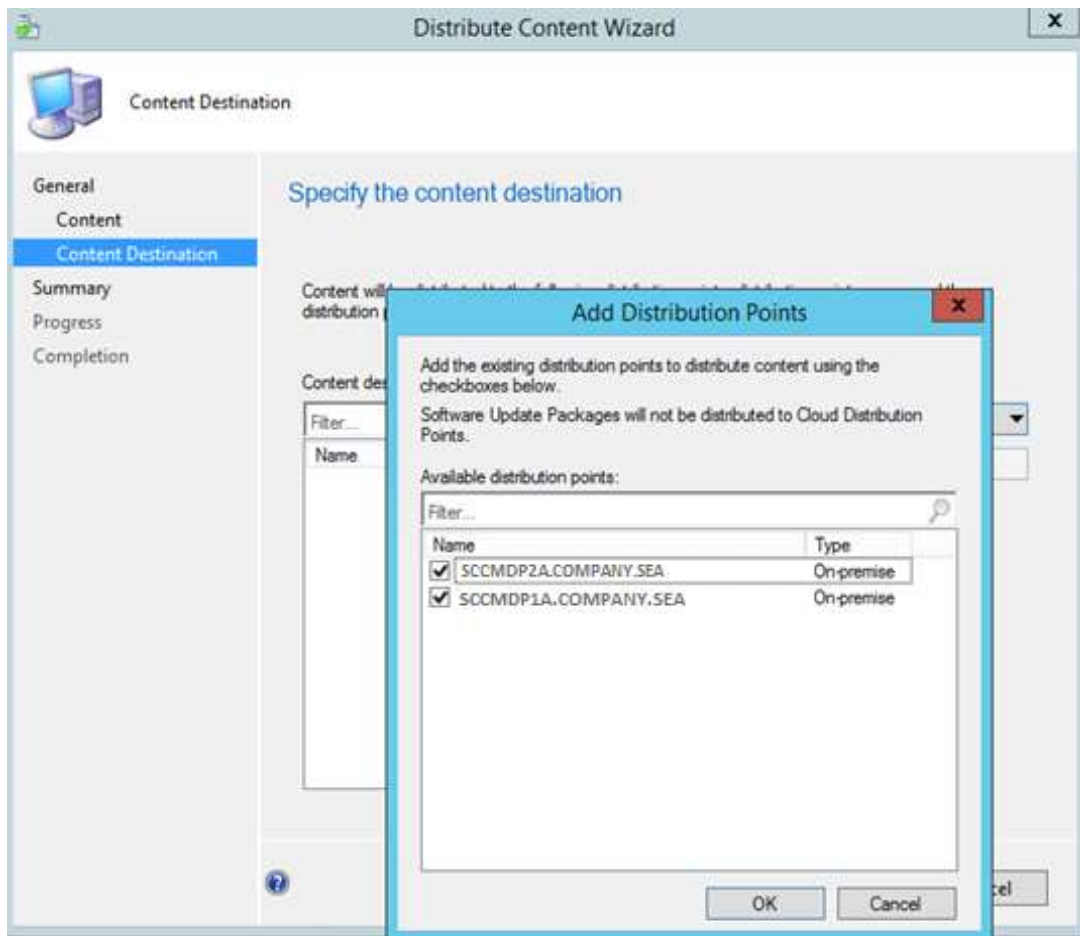


In the **Distribute Content Wizard**, do the following:

- Click **Next** past the first two screens of the wizard.
- Click the **Add** button on the *Specify the content destination* screen and select **Distribution Point**.



- Check both checkboxes on the **Add Distribution Points** page and click **OK** to close the window.

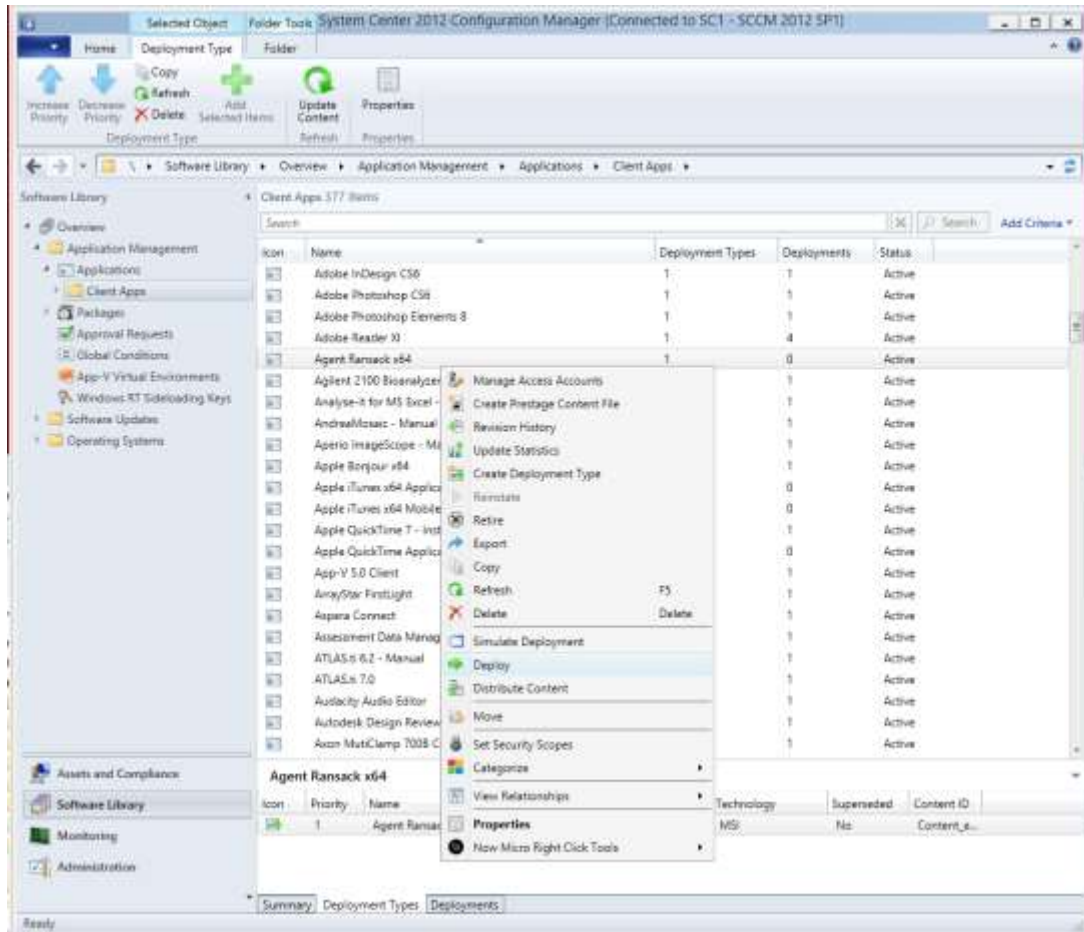


- Click **Next** twice and **Close** to exit the **Distribute Content Wizard**.

Find the application package in the SCCM Client Apps list and move on to the **5. How to Deploy a Package** section.

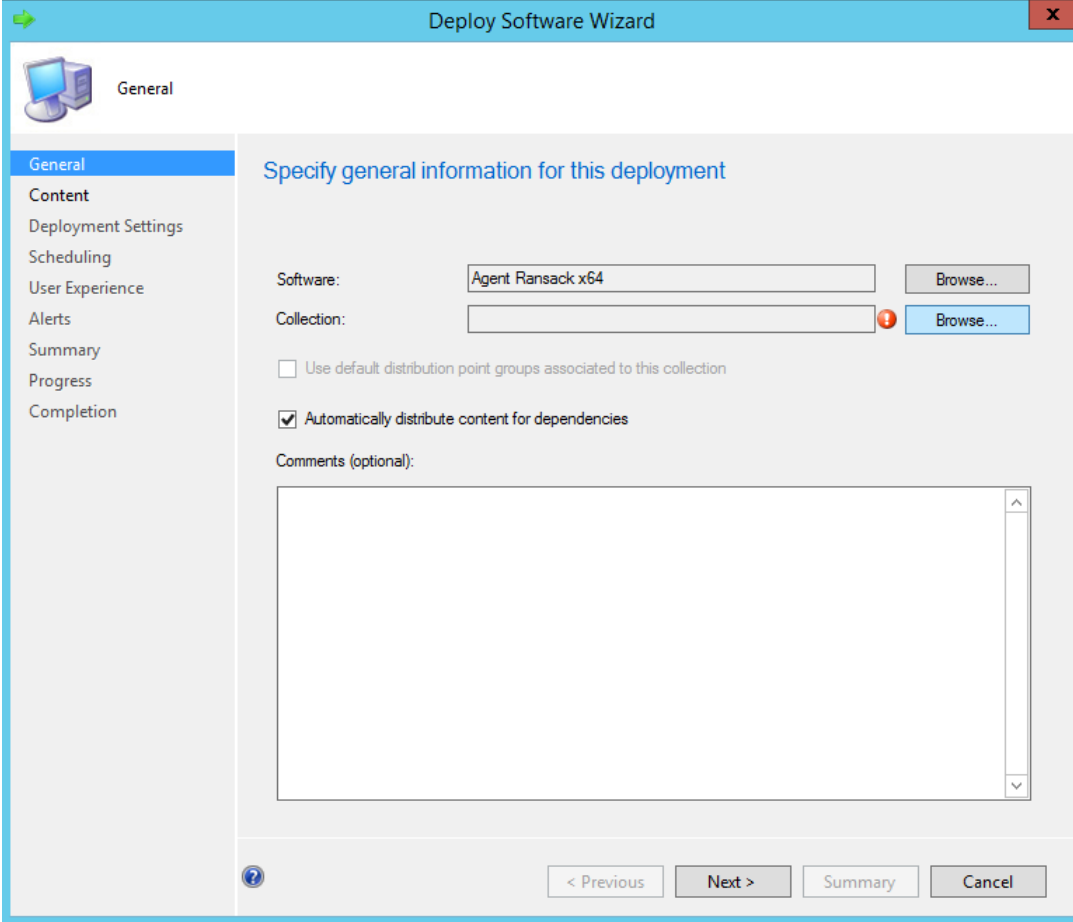
5. How to Deploy a Package

Find the application package in the SCCM **Client Apps** list, right-click the package name, and select **Deploy**.



In the **Deploy Software Wizard**, do the following:

- On the *Specify general information for this deployment* screen, click the **Browse...** button next to **Collection**:

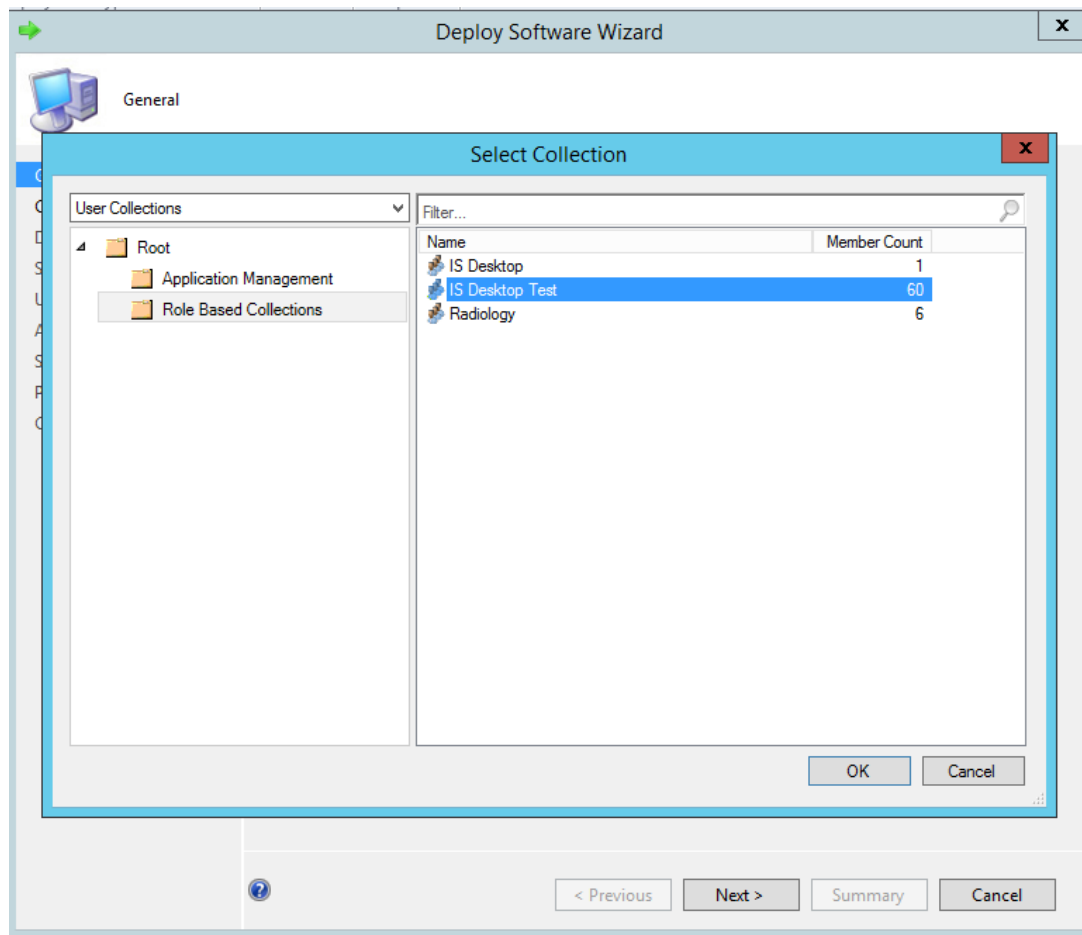


The screenshot shows the 'Deploy Software Wizard' window with the 'General' tab selected. The window title is 'Deploy Software Wizard'. On the left is a navigation pane with the following items: General (selected), Content, Deployment Settings, Scheduling, User Experience, Alerts, Summary, Progress, and Completion. The main area is titled 'Specify general information for this deployment'. It contains the following fields and controls:

- Software:** A text box containing 'Agent Ransack x64' and a 'Browse...' button.
- Collection:** An empty text box with a red warning icon to its right and a 'Browse...' button.
- ☐ Use default distribution point groups associated to this collection
- ☒ Automatically distribute content for dependencies
- Comments (optional):** A large text area for comments.

At the bottom of the window are four buttons: '< Previous', 'Next >', 'Summary', and 'Cancel'. A help icon (?) is located at the bottom left of the main area.

- On the **Select Collection** screen, select **Role Based Collections** under *User Collections*, then select **IS Desktop Test** in the right-hand pane, and click **OK** to close the window.



- Click **Next**.

- Wait for the *Specify the content destination* page to self-populate the distribution points list and then click **Next**.

Deploy Software Wizard

Content

Specify the content destination

Distribution points or distribution point groups that the content has been distributed to:

Name	Type
\\SCCMDP1A.COMPANY.SEA...	Distribution point
\\SCCMDP2A.COMPANY.SEA...	Distribution point

Additional distribution points, distribution point groups, and the distribution point groups that are currently associated with collections to distribute content to:

Filter...

Name	Description	Associations
There are no items to show in this view.		

< Previous **Next >** Summary Cancel

- Accept all the defaults on the following pages by clicking **Next** and then click **Close** to exit the wizard.

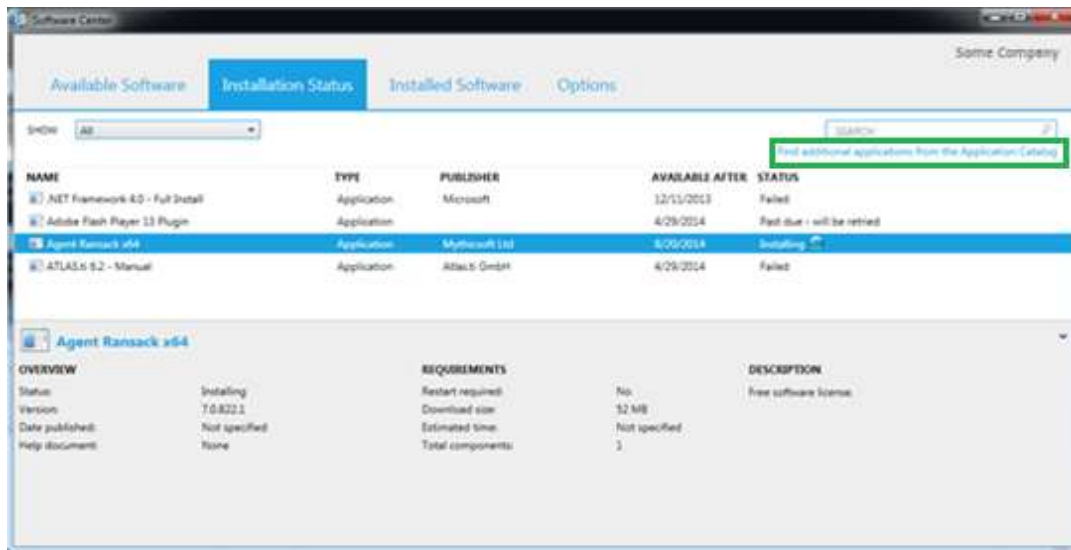
Be sure to wait 10-15 minutes for SCCM to complete processing before moving on to the next **6. Testing the Application Package** section.

6. Testing the Application Package

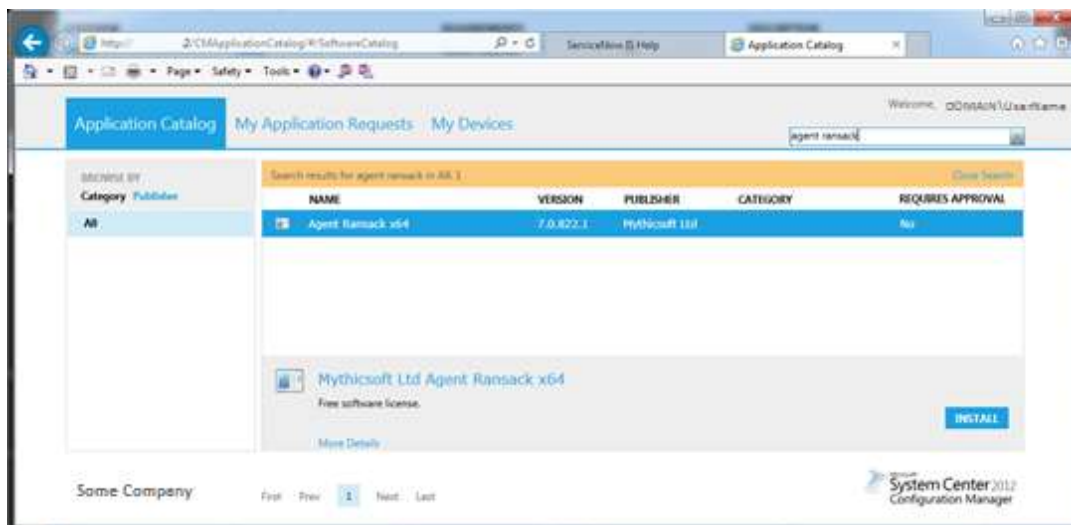
After waiting for the 10-15 minutes required for SCCM to copy the application contents from the Software Vault to the Distribution Points, test the SCCM application package install.

On a Windows 7 system, start **Software Center** which can be found in the **Start Menu** under *Microsoft System Center 2012, Configuration Manager*.

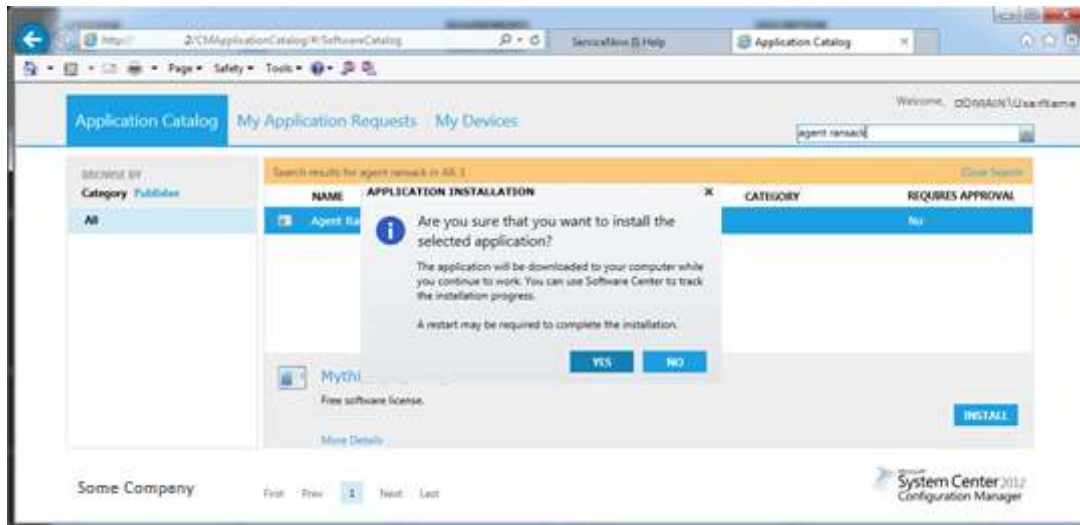
In **Software Center**, select the **Installation Status** tab, and click the *Find additional applications from the Application Catalog* link under the **SEARCH** text box.



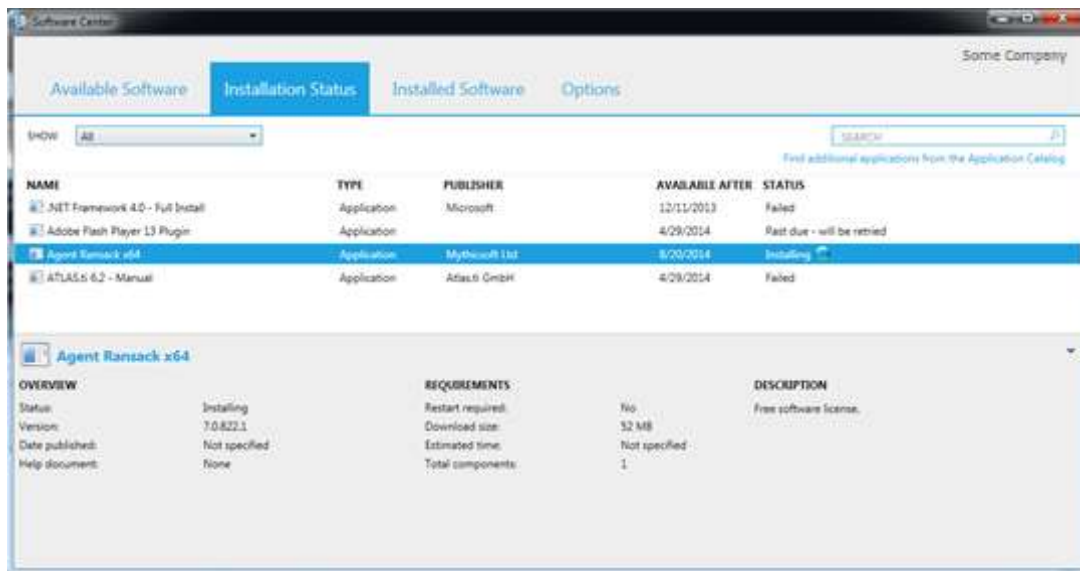
After Internet Explorer opens, type the application name in the *Search Application Catalog* text box and press **Enter**.



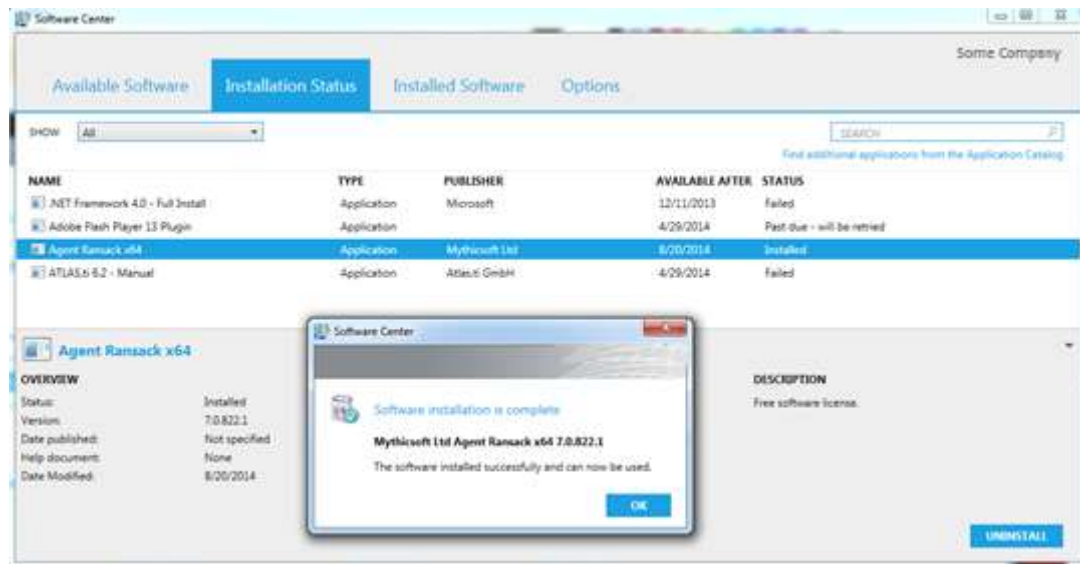
Click the **Install** button and click **Yes** on the *Application Installation* confirmation popup.



Switch back to the **Software Center** window to watch the application install.



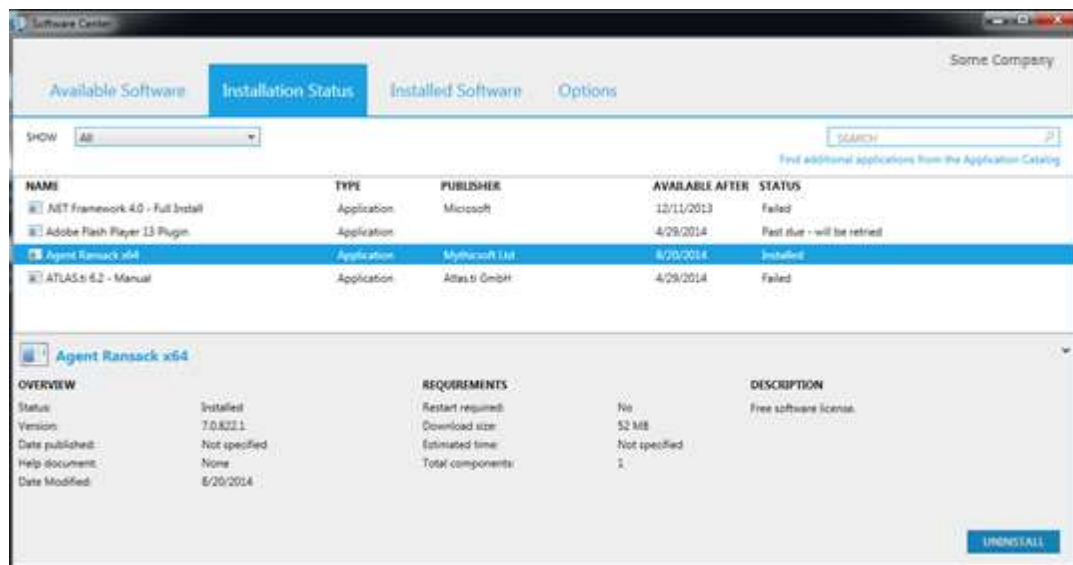
Click **OK** on the *Software installation is complete* popup message.



Start the application to ensure that it installed correctly.

If the application does create a shortcut in the **Start Menu** or on the user **Desktop**, then the installation method will need to be modified to a **Manual** method as the setup program requires user interaction to successfully create and set environmental variables.

In the **Software Center** window, select the application name and click the **Uninstall** button to ensure that the application is successfully removed from the local system.



The application should drop off the installed software list and a popup message indicating a **Removal Complete** should appear.

Install or uninstall errors need to be investigated to determine how the SCCM package needs to be modified.

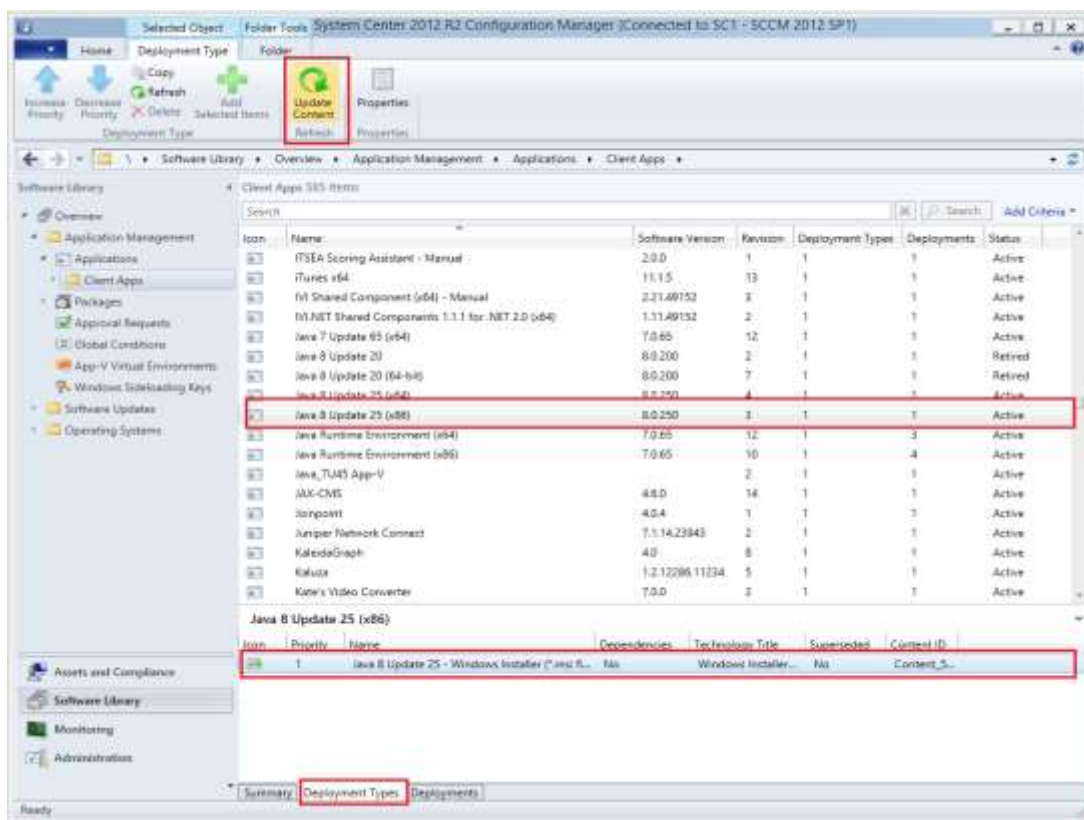
7. Re-Testing the SCCM Application Package

After modifying a package's settings or updating files on the Software Vault, the application package will need its content updated on the Distribution Servers. Then, before attempting to install through Software Center, the Configuration Manager cache needs to be cleared on the application packaging desktop test system.

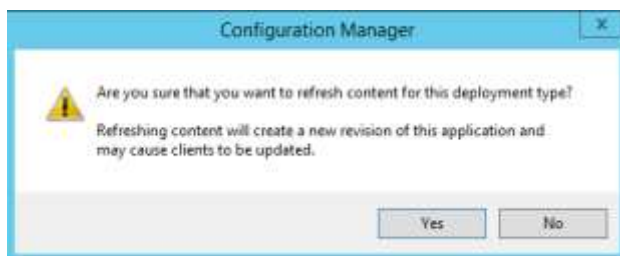
Updating Content on the Distribution Server

On the SCCM Configuration Manager console, do the following:

- Select the **Software Library**, expand **Applications**, and then expand **Client Apps**.
- Find the application package in the **Client Apps** list and select the application package.
- Select the **Deployment Types** tab at the bottom of the console window and select the installer name in the lower pane.
- Click the **Update Content** button below the menu bar at the top of the console window.



- Click **Yes** on the refresh content confirmation popup message.



Verifying the Content Update on the Distribution Server

On the SCCM Configuration Manager console, do the following:

- Verify that the package is updating by selecting the **Summary** tab at the bottom of the console window and scrolling down to the **Content Status** section to view the content update status.
- Color key:
 - **Green** indicates a successful content update.
 - **Yellow** means the content update is in progress.
 - **Red** indicates a failed content update that needs to be investigated.
 - Make sure there are no open files in the application's **Software Vault** folder (e.g., an uninstall program open for editing in PowerShell or a ReadMe file in Notepad).

Java 8 Update 25 (x86) - Content Update in progress...

System Center 2012 R2 Configuration Manager (Connected to SC1 - SCCM 2012 SP1)

Software Library > Overview > Application Management > Applications > Client Apps

Icon	Name	Software Version	Revised	Deployment Types	Deployments	Status
	Java 8 Update 20 (64-bit)	8.0.200	7	1	1	Retired
	Java 8 Update 25 (x64)	8.0.250	4	1	1	Active
	Java 8 Update 25 (x86)	8.0.250	4	1	1	Active
	Java Runtime Environment (x64)	7.0.65	12	1	3	Active
	Java Runtime Environment (x86)	7.0.65	10	1	4	Active
	Java_TU45 App-V		2	1	1	Active
	IOX-CMS	4.0.0	14	1	1	Active
	Isopoint	4.0.4	1	1	1	Active

Java 8 Update 25 (x86)

Application Properties

Software Version: 8.0.250
Manufacturer: Oracle Corporation
Superseded: No
Comments: Free software license

Application Statistics

Devices with Application: 1
Devices with Installation: 0
Failure: 0
Users with Application: 1
Users with Installation: 0
Users with Catalog: 3
Installations: 3
Last Update: 1/13/2015 4:39 PM

Related Objects

[Content Status](#)

Application Status

Revision: 4
Status: Active
Deployments: 1

Content Status

2 Targeted (Last Update: 1/14/2015 11:59 AM)

Summary | Deployment Types | Deployments

Java 8 Update 25 (x86) - Content Update success!

The screenshot shows the System Center 2012 R2 Configuration Manager console. The left pane displays the navigation tree with 'Software Library' selected. The main pane shows the 'Client Apps' view for 'Java 8 Update 25 (x86)'. A table lists various software items, including Java 7 Update 65 (x64), Java 8 Update 20, Java 8 Update 25 (x64), and Java 8 Update 25 (x86). The 'Java 8 Update 25 (x86)' item is highlighted, and its details are shown below. The 'Application Properties' section includes software version (8.0.250), manufacturer (Oracle Corporation), and comments (Free software license). The 'Application Statistics' section shows 1 device with the application, 0 failures, and 1 user with the application. The 'Content Status' section shows a green circle indicating 2 targeted devices, with a last update of 1/14/2015 12:01 PM. The 'Summary' tab at the bottom shows 4 revisions, 1 active status, and 1 deployment.

Icon	Name	Software Version	Revision	Deployment Types	Deployments	Status
	VM.NET Shared Components 1.1.1 for .NET 2.0 (x64)	1.11.49152	2	1	1	Active
	Java 7 Update 65 (x64)	7.0.65	12	1	1	Active
	Java 8 Update 20	8.0.209	2	1	1	Retired
	Java 8 Update 20 (64-bit)	8.0.209	7	1	1	Retired
	Java 8 Update 25 (x64)	8.0.250	4	1	1	Active
	Java 8 Update 25 (x86)	8.0.250	4	1	1	Active
	Java Runtime Environment (x64)	7.0.65	12	1	3	Active
	Java Runtime Environment (x86)	7.0.65	10	1	4	Active

Java 8 Update 25 (x86)

Application Properties

Software Version: 8.0.250
Manufacturer: Oracle Corporation
Superseded: No
Comments: Free software license.

Application Statistics

Devices with Application: 1
Devices with Installation: 0
Failure: 0
Users with Application: 1
Users with Installation: 0
Failure: 0
Users with Catalog: 3
Installations: 3
Last Update: 1/13/2015 4:39 PM

Content Status

2 Targeted (Last Update: 1/14/2015 12:01 PM)

Legend: Success: 2, In Progress: 0, Failed: 0, Unknown: 0

Application Status

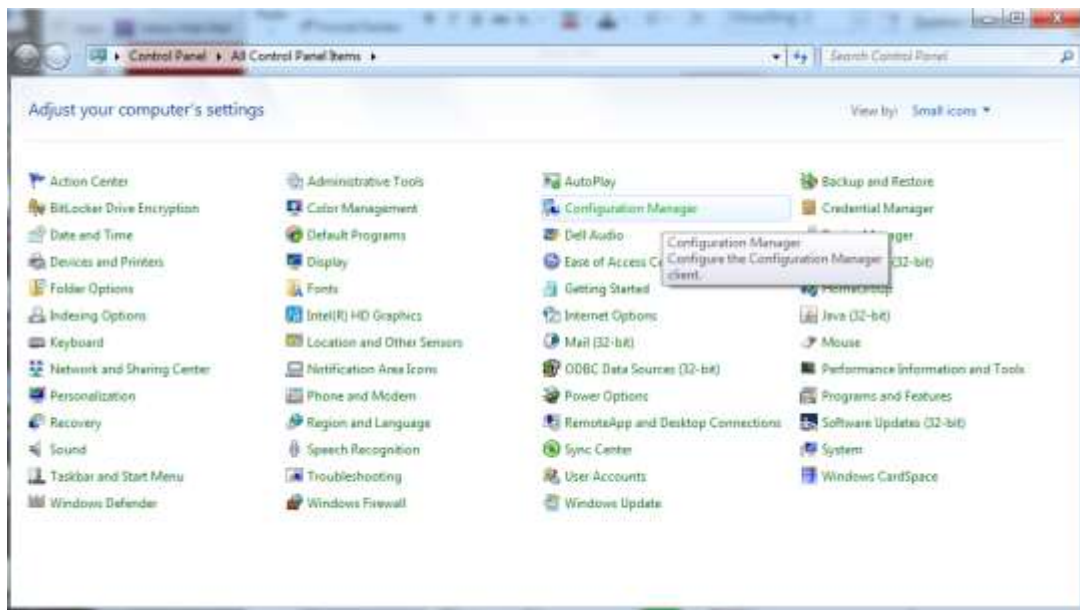
Revision: 4
Status: Active
Deployments: 1

Summary: Deployment Types | Deployments

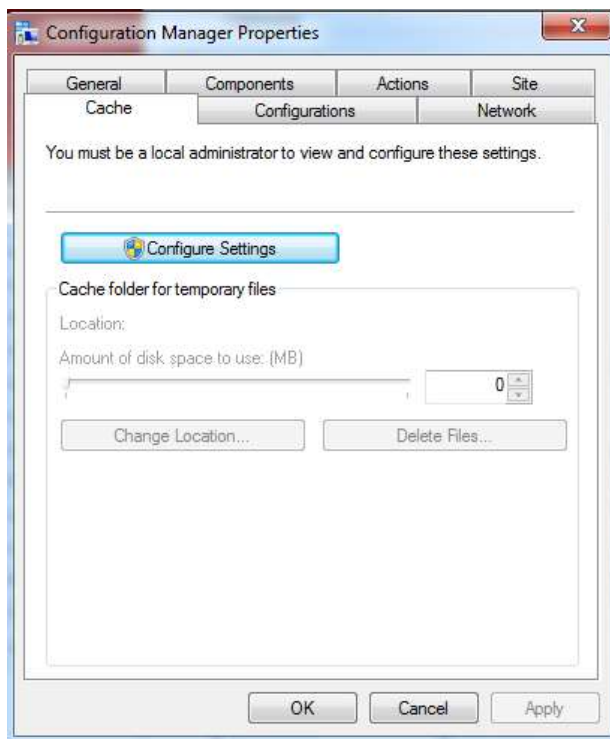
Clearing the Configuration Manager Cache

On the application packaging desktop test system:

- Open the **Configuration Manager** control panel applet.



- Click the **Cache** tab in the *Configuration Manager Properties* window.



- Click the **Configure Settings** button and then the **Delete Files** button.

- Answer **Yes** on the *Confirm Delete Files* window.
 - **DO NOT** enable the *Delete persisted cache content* checkbox.



- Close the *Configuration Manager Properties* window.

Go to the top of this section and repeat the steps in the **6. Testing the Application Package** section.

Program Visibility Options

Issue: An application installs silently but prompts the user for confirmation during the uninstall process.

The resulting application's SCCM package *User Experience* settings would be:

- Installation behavior: **Install for user**
- Logon requirement: **Whether or not a user is logged on**
- Installation program visibility: **Normal**

The program's visibility is set to **Normal** so that the uninstall confirmation pop-up message is displayed so the technician can acknowledge it and complete the uninstall process.

If visibility is set to **Hidden**, the uninstall confirmation pop-up message does not appear for the user to acknowledge and the uninstall never completes making it appear to hang.

If this happens, open **Task Manager** and end the uninstall process so that SCCM fails. Open the package properties, change the visibility to **Normal**, and update the content. Clear the target system's SCCM cache and try uninstalling the application again. The uninstall confirmation pop-up message will appear for the technician to acknowledge and the uninstall will complete successfully.

The SCCM Application Packaging how-to documentation continues in:

SCCM Application Packaging – Part 2 – PowerShell