

App management documentation

Use Configuration Manager to manage and deploy applications, scripts, and packages. Automate the deployments, or let users install apps from Software Center.

About app management

OVERVIEW

[Introduction to app management](#)

[Plan for app management](#)

[Plan for Software Center](#)

TUTORIAL

[Create and deploy an app](#)

Get started

HOW-TO GUIDE

[Create apps](#)

[Create and run PowerShell scripts](#)

DEPLOY

[Deploy apps](#)

[Deploy and update Microsoft Edge](#)

Top tasks

HOW-TO GUIDE

[Create app groups](#)

[Approve apps](#)

[Install apps for a device](#)

[User device affinity](#)

[Monitor app usage with software metering](#)

 **REFERENCE**

[Troubleshoot app deployments](#)

[Common error codes for app installation](#)

[Packages and programs](#)

Introduction to application management in Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

In this article, you'll learn the basics before you start working with Configuration Manager applications.

💡 Tip

If you're already familiar with how to manage applications in Configuration Manager, skip this article. Move on to creating a sample application: [Create and deploy an application](#).

What is an application?

Although *application* or *app* is a widely used term in computing, in Configuration Manager, it means something specific. Think of an application like a box. This box contains one or more sets of installation files for a software package (known as a *deployment type*), plus instructions on how to deploy the software.

When you deploy the application to devices, **requirements** decide which deployment type Configuration Manager installs on the device.

You can do many more things with an application. You'll learn about these things as you read this guide. The following sections introduce some concepts you'll need to know before you start to dig deeper:

Deployment type

If the *application* is the box, then the *deployment type* is the set of contents in the box. An application needs at least one deployment type, as it determines how to install the app. Use more than one deployment type to configure different content and installation program for the same application.

For example, your company has a line-of-business application called Astoria. The application developers provide the following ways of installing the app:

- Windows Installer package for full functionality on Windows 10 devices

- An App-V package for use in the terminal server farm
- An web app for mobile users

You create a single application for Astoria in Configuration Manager. The application defines the high-level metadata about the app that's common across all installation methods and platforms. You then create three deployment types for the available installation methods, and deploy the application to all users. Based on the requirements and other configurations on the deployment types, Configuration Manager determines the right method in each use case.

For more information, see [Create deployment types for the application](#).

Requirements

In previous versions of Configuration Manager, you would create a collection of devices to deploy an application to. Although you can still create a collection, use *requirements* to specify more detailed criteria for an application deployment.

For example, specify that an application can only install on devices that run Windows 10. When you deploy the application to all of your devices, it only installs on devices that run Windows 10.

Configuration Manager evaluates requirements to determine whether it installs an application and any of its deployment types. Then it determines the correct deployment type by which to install an application. Every seven days, by default, the Configuration Manager client reevaluates requirement rules to determine compliance according to the client setting **Schedule re-evaluation for deployments**.

For more information, see [Create and deploy an application](#) and [Deployment type Requirements](#).

Global conditions

While you use requirements with a specific deployment type in a single application, you can also create *global conditions*. These conditions are a library of predefined requirements that you can use with any application and deployment type. Configuration Manager includes a set of built-in global conditions, or you can create your own.

For more information, see [Create global conditions](#).

Simulated deployment

A *simulated deployment* evaluates the requirements, detection method, and dependencies for an application. A client reports the results without actually installing the application.

For more information, see [Simulate application deployments](#).

Deployment action

A *deployment action* specifies whether you want to install or uninstall the application you're deploying. Not all deployment types support the uninstall action.

For more information, see [Deploy applications](#).

Deployment purpose

The *deployment purpose* specifies whether the deployment app is **Required** or **Available**:

- The client automatically installs a *required* deployment according to the schedule that you set. If the application isn't hidden, a user can track its deployment status. They can also use Software Center to install the application before the deadline.
- If you deploy the application to a user as *available*, they see it in Software Center, and can request it on demand.

For more information, see [Deploy applications](#).

Revisions

When you make *revisions* to an application or a deployment type, Configuration Manager creates a new version of the application. Take the following actions in the Configuration Manager console:

- Display the history of each application revision
- View its properties
- Restore a previous version of an application
- Delete an old version

For more information, see [Revise applications](#).

Detection method

Use *detection methods* to discover whether a device has already installed an application. If the detection method indicates the application is installed, Configuration Manager

doesn't attempt to install it again.

For more information, see [Deployment type Detection Method options](#).

Dependencies

Dependencies define one or more deployment types from another application that the client must install before it installs this deployment type.

For more information, see [Deployment type Dependencies](#).

Supersedence

Configuration Manager lets you upgrade or replace existing applications by using a *supersedence* relationship. When you supersede an application, you specify a new deployment type to replace the deployment type of the superseded application. You can also decide whether to upgrade or uninstall the superseded application before the client installs the superseding application.

For more information, see [Application supersedence](#).

User-centric management

Configuration Manager applications support *user-centric management*, which lets you associate specific users with specific devices. Instead of having to remember the name of a user's device, deploy apps to the user and to the device. This functionality helps you make sure the most important apps are always available on each of the user's devices. If a user acquires a new computer, Configuration Manager automatically installs their apps on the device before they sign in.

For more information, see [Link users and devices with user device affinity](#).

Application group

Create a group of applications that you can send to a user or device collection as a single deployment. The metadata you specify about the app group is seen in Software Center as a single entity. You can order the apps in the group so that the client installs them in a specific order.

For more information, see [Create application groups](#).

What application types can you deploy?

Configuration Manager lets you deploy the following app types:

- Windows Installer (msi)
- Windows app package and app bundles (appx, appxbundle, msix, msixbundle)
- Windows app package in the Microsoft Store
- Script installer for third-party installers and script wrappers
- Microsoft App-V v4 and v5
- macOS
- A non-OS deployment task sequence for complex apps

Additionally, when you manage devices through Configuration Manager [on-premises device management](#), manage these further app types:

- Windows Phone app package (xap)
- Windows Phone app package in the Microsoft Store
- Windows Installer through MDM (msi)
- Web application

State-based applications

Configuration Manager applications use state-based monitoring. You can track the last application deployment state for users and devices. The state messages display information about individual devices. For example, if you deploy an application to a collection of users, you can view the compliance state of the deployment and the deployment purpose in the Configuration Manager console. Monitor the deployment of all software from the **Monitoring** workspace in the Configuration Manager console. For more information, see [Monitor applications](#).

The Configuration Manager client regularly reevaluates application deployments. For example:

- A user uninstalls a deployed application. At the next evaluation cycle, Configuration Manager detects that the app isn't present. The client then automatically reinstalls the app.

- Configuration Manager didn't install an application on a device because it failed to meet the requirements. Later, a change is made to the device and it now meets the requirements. Configuration Manager detects this change, and the client installs the application.

You can set the re-evaluation interval for application deployments. Use the **Schedule re-evaluation for deployments** client setting in the **Software Deployment** group. For more information, see [About client settings](#).

Get started creating an application

If you want to jump right in and create an application, you'll find a walkthrough in the [Create and deploy an application](#) article.

If you're familiar with the basics and looking for more detailed reference information about all the available options, start to [Create applications](#).

Software Center

Software Center is a Windows application installed with the Configuration Manager client. Use it for the following actions:

- Browse for and request applications deployed to the device or the user
- Install and schedule software installations
- View installation status for applications, software updates, and operating systems
- Configure remote control settings
- Set up power management

For more information, see the following articles:

- [Plan for and configure application management](#)
- [Plan for Software Center](#)
- [Software Center user guide](#)

Packages and programs

Configuration Manager continues to support packages and programs that were used in previous versions of the product.

For more information, see [Packages and programs](#).

Next steps

Now that you understand the basic concepts of application management in Configuration Manager, continue to the following articles:

- [Create and deploy an example application](#)
- [Plan for and configure application management](#)
- [Create applications](#)

Create and deploy an application with Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

In this article, you'll learn how to create an application with Configuration Manager. In this example, you'll create and deploy the [CMPivot standalone](#) installer. For the purposes of this exercise, you'll configure it to only install on devices that are running Windows 11. Along the way, you'll learn about many of the things you can do to manage applications effectively.

Tip

The CMPivot standalone source file is in the Configuration Manager installation media or on the site server in the CD.Latest folder. Find it in the following folder:

`\SMSSETUP\TOOLS\CMPivot\CMPivot.msi`

This procedure is designed to give you an overview of how to create and deploy Configuration Manager applications. However, it doesn't cover all the configuration options, or how to create and deploy applications for other platforms.

For specific details that are relevant to each platform, see one of the following articles:

- [Create Windows applications](#)
- [Create Windows Phone applications](#)
- [Create Mac computer applications](#)
- [Create Windows Embedded applications](#)

If you're already familiar with Configuration Manager applications, you can skip this article. To learn about all the options that are available when you create and deploy applications, see [Create applications](#).

Before you start

Make sure that you've reviewed the information in [Introduction to application management](#). That article helps you prepare your site to install applications and understand the terminology that's used here.

Make sure that the installation files for the CMPivot standalone app are in an accessible location on your network. This example uses the following path:

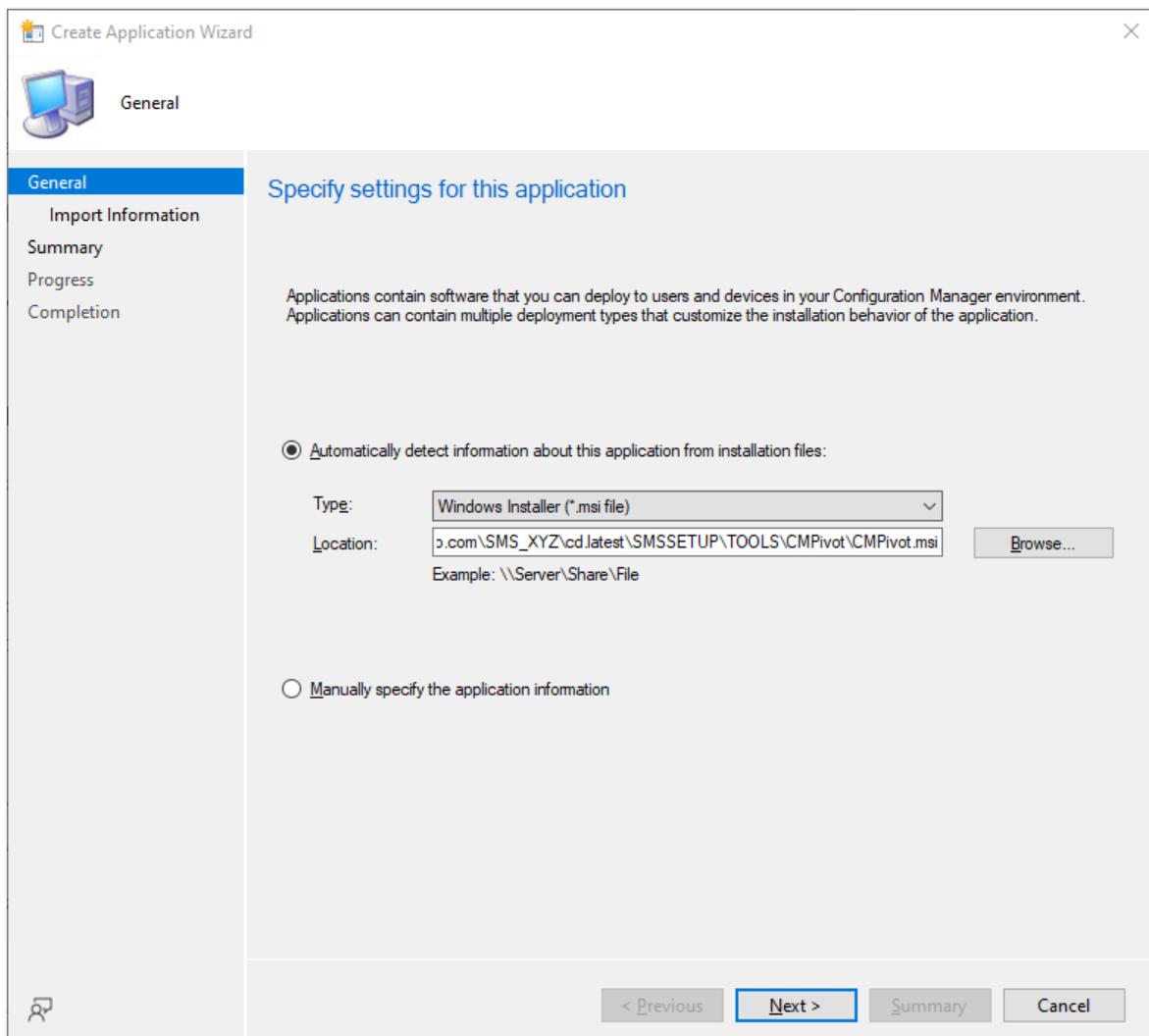
```
\cm01.contoso.com\SMS_XYZ\cd.latest\SMSSETUP\TOOLS\CMPivot\CMPivot.msi
```

Create the application

Use the following procedure to start the Create Application Wizard and create the application:

1. In the Configuration Manager console, choose **Software Library > Application Management > Applications**.
2. On the **Home** tab, in the **Create** group, choose **Create Application**.
3. On the **General** page of the **Create Application Wizard**, choose **Automatically detect information about this application from installation files**. This action pre-populates some of the information in the wizard with information that's extracted from the installation .msi file. Then specify the following information:
 - **Type:** Choose **Windows Installer (*.msi file)**.
 - **Location:** Select **Browse** to choose the location of the installation file **CMPivot.msi**. Make sure the location is specified in the form `\Server\Share\File.msi` for Configuration Manager to locate the installation files.

You'll end up with something that looks like the following screenshot:



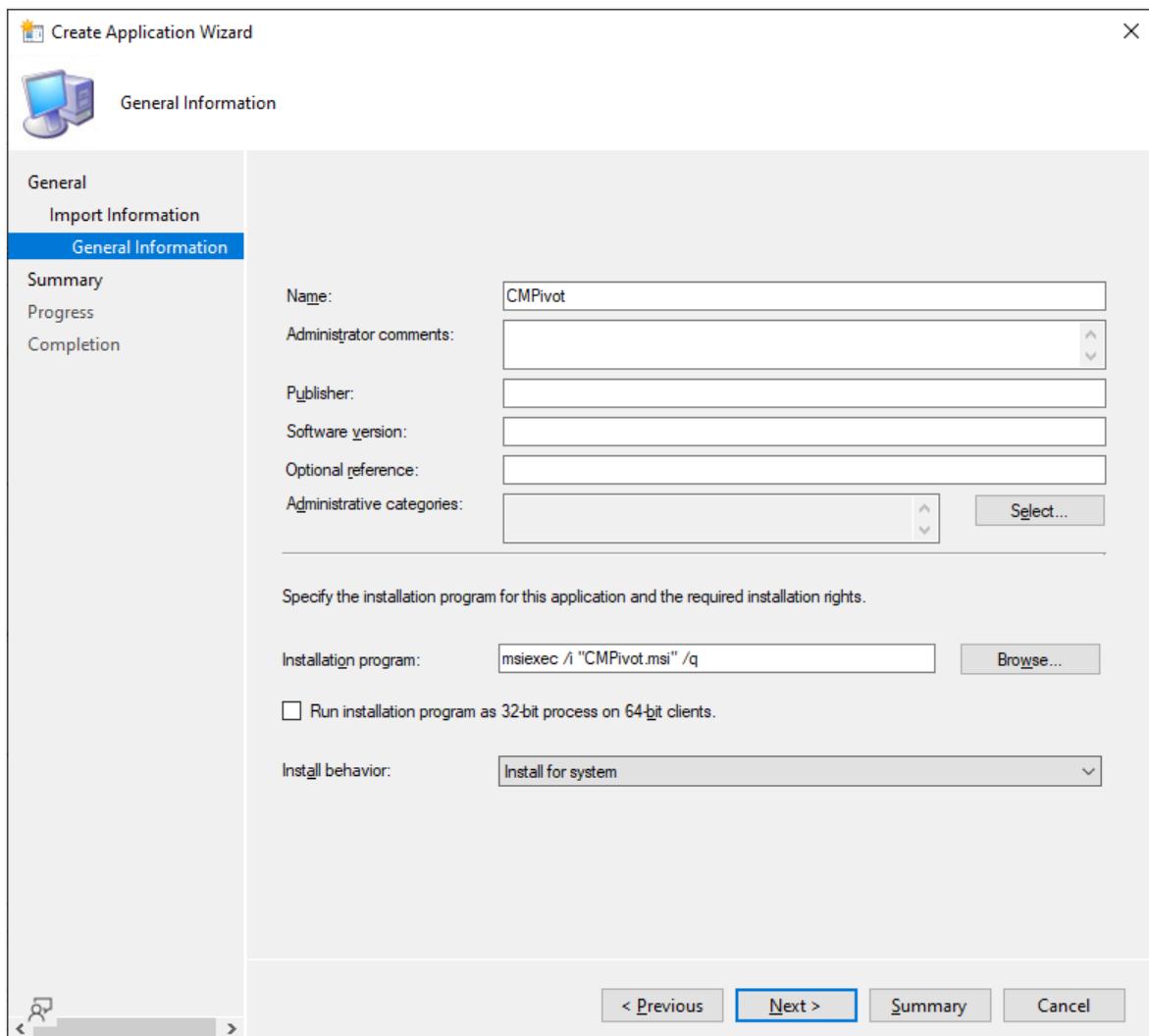
4. Choose **Next**. On the **Import Information** page, you'll see some information about the app and any associated files that were imported to Configuration Manager. Once you're done, choose **Next** again.
5. On the **General Information** page, you can supply further information about the application to help you sort and locate it in the Configuration Manager console.

The **Installation program** field lets you specify the full command line that will be used to install the application on PCs. You can edit this field to add your own properties. For example, `/q` for an unattended installation.

Tip

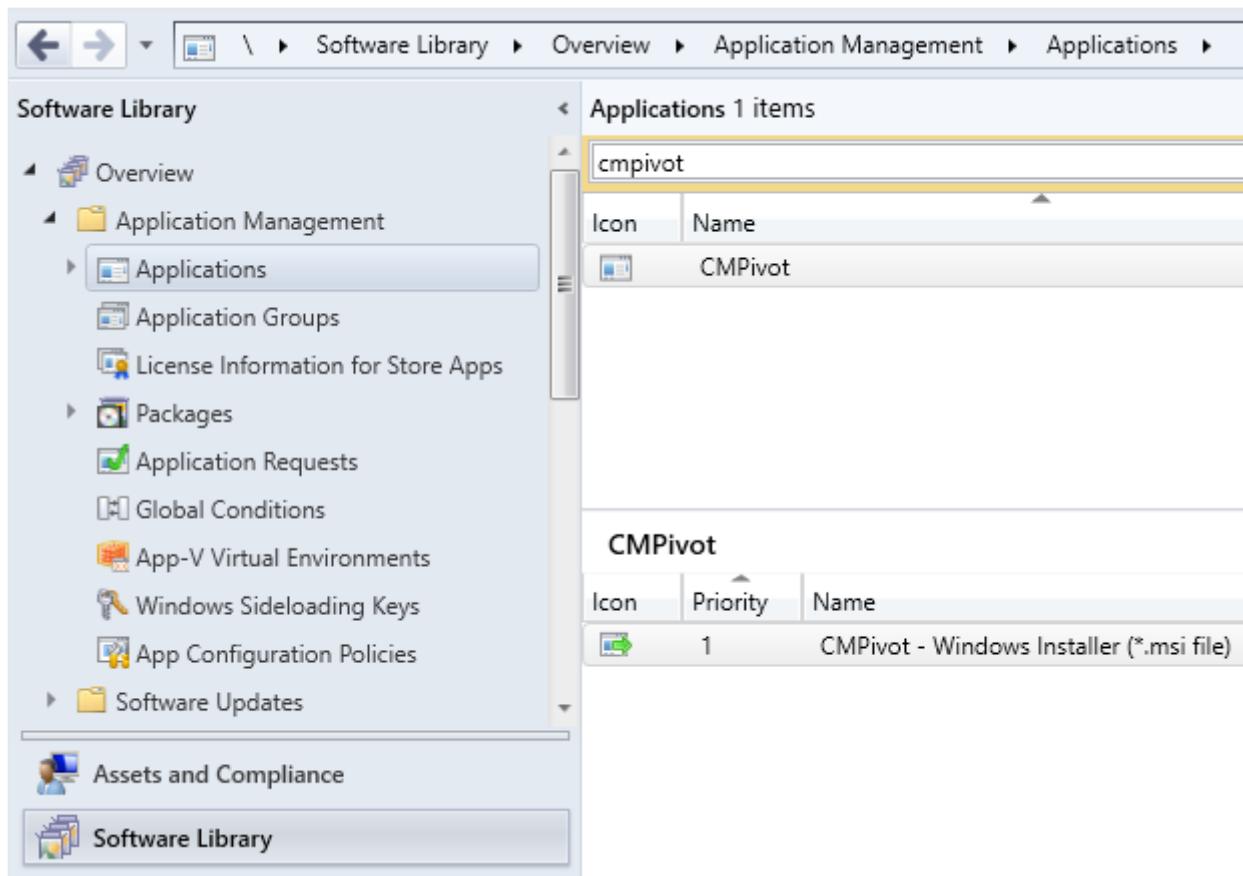
Some of the fields on this page of the wizard might have been filled in automatically when you imported the application installation files.

You'll end up with a screen that looks similar to the following screenshot:



6. Choose **Next**. On the Summary page, you can confirm your application settings and then complete the wizard.

You've finished creating the app. To find it, in the **Software Library** workspace, expand **Application Management**, and then choose **Applications**. For this example, you'll see:



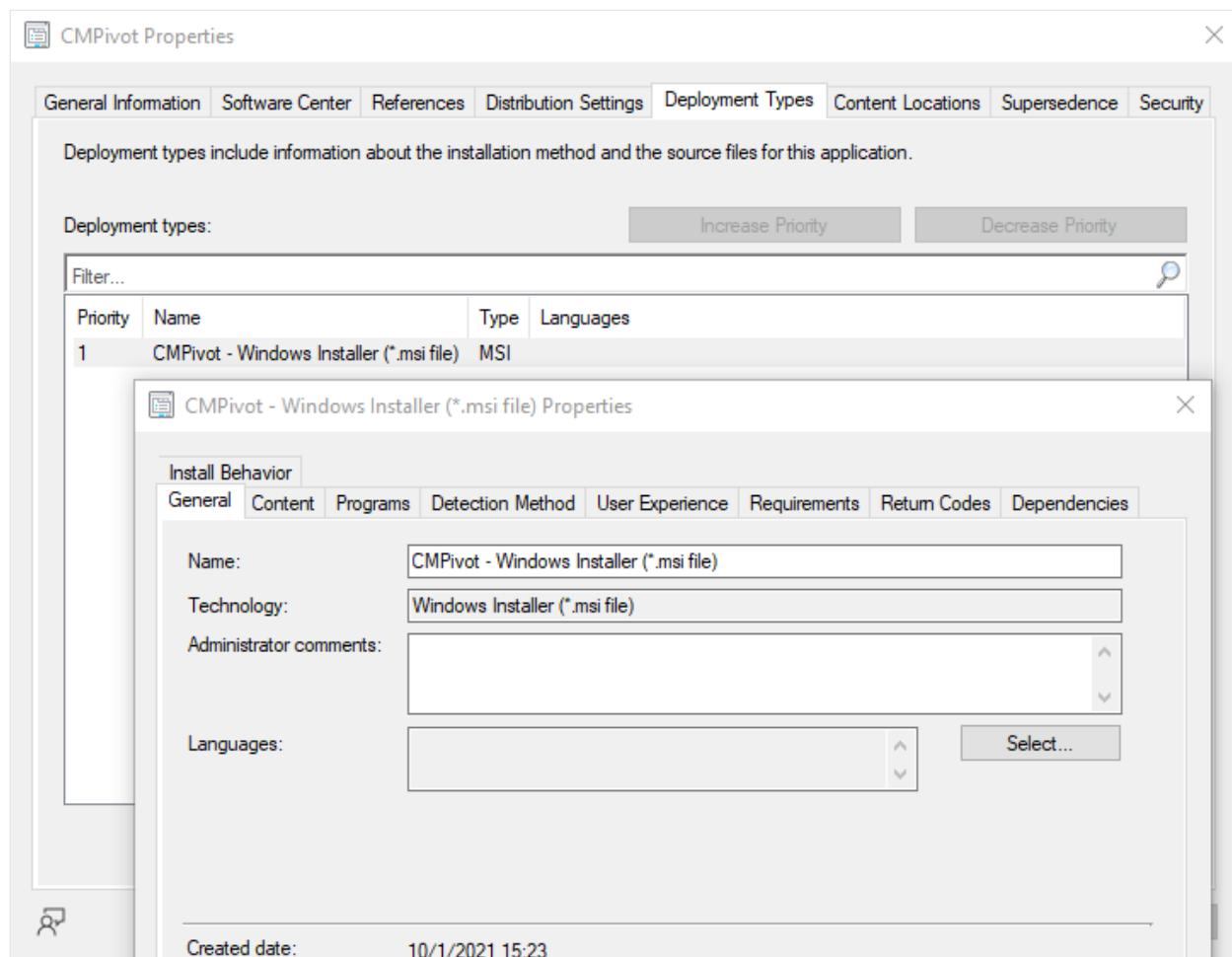
Examine the properties

Now that you've created an application, you can refine the application settings if you need to. To look at the application properties, select the app, and then, in the **Home** tab in the **Properties** group, choose **Properties**.

In the **CMPivot Properties** dialog box, you'll see many items that you can configure to refine the behavior of the application. For more information about all the settings you can configure, see [Create applications](#).

For the purposes of this example, you'll just be changing some properties of the application's deployment type. In the app properties window, switch to the **Deployment Types** tab. Select the **CMPivot - Windows Installer (*.msi file)** deployment type, and then select **Edit**.

You'll see a dialog box like this one:

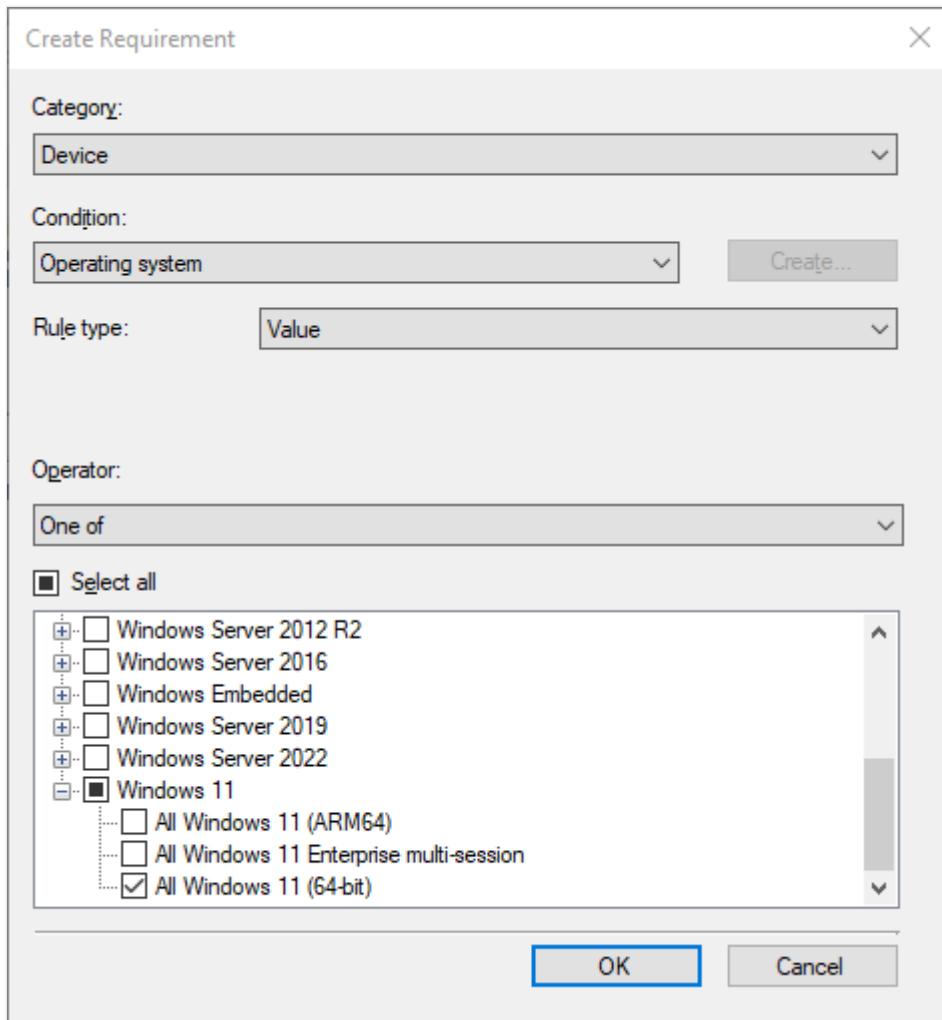


Add a requirement

Requirements specify conditions that must be met before an application is installed on a device. You can choose from built-in requirements or you can create your own. In this example, you add a requirement that the application will only get installed on devices that are running Windows 11.

1. On the deployment type properties page, switch to the Requirements tab.
2. Select **Add** to open the **Create Requirement** window. Specify the following information:
 - Category: Device
 - Condition: Operating system
 - Rule type: Value
 - Operator: One of
 - From the OS list, select All Windows 11 (64-bit).

You'll end up with a dialog box that looks like this:



3. Select **OK** to close each property page that you opened. Then return to the **Applications** list in the Configuration Manager console.

💡 Tip

Requirements can help reduce the number of Configuration Manager collections you need. Because you just specified that the application can only get installed on devices that are running Windows 11, you can later deploy this to a collection that contains PCs that run many different operating systems. But the application will only get installed on Windows 11 devices.

Distribute the application content

Next, to deploy the application to PCs, make sure that the application content is copied to a distribution point. PCs access the distribution point to install the application.

💡 Tip

To find out more about distribution points and content management in Configuration Manager, see [Manage content and content infrastructure](#).

1. In the Configuration Manager console, choose **Software Library**.
2. In the **Software Library** workspace, expand **Applications**. Then, in the list of applications, select the **CMPivot** that you created.
3. On the **Home** tab, in the **Deployment** group, choose **Distribute Content**.
4. On the **General** page of the **Distribute Content Wizard**, check that the application name is correct, and then choose **Next**.
5. On the **Content** page, review the information that will be copied to the distribution point, and then choose **Next**.
6. On the **Content Destination** page, choose **Add** to select one or more distribution points, or distribution point groups on which to install the application content.
7. Complete the wizard.

You can check that the application content was copied successfully to the distribution point from the **Monitoring** workspace, under **Distribution Status > Content Status**.

Deploy the application

Next, deploy the application to a device collection in your hierarchy. In this example, you deploy the application to the **All Systems** device collection.

Tip

Remember that only Windows 11 computers will install the application because of the requirements that you selected earlier.

1. In the Configuration Manager console, choose **Software Library > Application Management > Applications**.
2. From the list of applications, select the application that you created earlier (**CMPivot**), and then, on the **Home** tab in the **Deployment** group, choose **Deploy**.
3. On the **General** page of the **Deploy Software Wizard**, choose **Browse** to select the **All Systems** device collection.

4. On the **Content** page, check that the distribution point from which you want PCs to install the application is selected.
5. On the **Deployment Settings** page, make sure that the deployment action is set to **Install**, and the deployment purpose is set to **Required**.

 **Tip**

By setting the deployment purpose to **Required**, you make sure that the application is installed on PCs that meet the requirements that you set. If you set this value to **Available**, then users can install the application on demand from Software Center.

6. On the **Scheduling** page, you can configure when the application will be installed. For this example, select **As soon as possible after the available time**.
7. On the **User Experience** page, choose **Next** to accept the default values.
8. Complete the wizard.

Use the information in the following **Monitor the application** section to see the status of your application deployment.

Monitor the application

In this section, you'll take a quick look at the deployment status of the application that you deployed.

1. In the Configuration Manager console, choose **Monitoring > Deployments**.
2. From the list of deployments, select **CMPivot**.
3. On the **Home** tab, in the **Deployment** group, choose **View Status**.
4. Select one of the following tabs to see more status updates about the application deployment:
 - **Success:** The application installed successfully on the indicated PCs.
 - **In Progress:** The application is still installing.
 - **Error:** An error occurred installing the application on the indicated PCs. Further information about the error is also displayed.

- **Requirements Not Met:** No installation attempt was made on the indicated devices because they didn't meet the requirements you configured. In this example, because they don't run on Windows 11.
- **Unknown:** Configuration Manager was unable to report the status of the deployment. Check back again later.

💡 Tip

There are a few ways you can monitor application deployments. For more information, see [Monitor applications](#).

User experience

Users who have PCs that are managed by Configuration Manager and running Windows 11 see a message telling them that they must install the CMPivot application. Once they accept the deployment, the application gets installed.

Next steps

[User notifications](#)

Plan for and configure application management in Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Use the information in this article to help you implement the necessary dependencies to deploy applications in Configuration Manager.

Dependencies external to Configuration Manager

Internet Information Services (IIS)

IIS is required on the servers that run the following site system roles:

- Management point
- Distribution point

For more information, see [Site and site system prerequisites](#).

Certificates on code-signed applications for mobile devices

When you code-sign applications to deploy them to mobile devices, don't use a certificate that was generated by using a Version 3 template (**Windows Server 2008, Enterprise Edition**). This certificate template creates a certificate that's incompatible with Configuration Manager applications for mobile devices.

If you use Active Directory Certificate Services to code-sign applications for mobile device applications, don't use a Version 3 certificate template.

Audit sign-in events for user device affinity

If you want to automatically create user device affinities, configure clients to audit sign-in events.

To determine automatic user device affinities, the Configuration Manager client reads sign-in events of type **Success** from the Windows security event log. Enable these events

with the following two audit policies:

- **Audit account logon events**
- **Audit logon events**

To automatically create relationships between users and devices, make sure that these two settings are enabled on client computers. You can use Windows Group Policy to configure these settings.

For more information on user device affinity, see [Link users and devices with user device affinity](#).

Configuration Manager dependencies

Management point

Clients contact a management point to download client policy and to locate content. Software Center uses the same management point for user-available application deployments.

Distribution point

Before you can deploy applications to clients, you need at least one distribution point in the hierarchy. By default, the site server has a distribution point site role enabled during a standard installation. The number and location of distribution points vary according to the specific requirements of your environment.

For more information about how to install distribution points and manage content, see [Manage content and content infrastructure](#).

Reporting services point

To use the reports in Configuration Manager for application management, first install and configure a reporting services point.

For more information, see [Introduction to reporting](#).

Client settings

Many client settings control how the client installs applications and the user experience on the device. These client settings include the following groups:

- Computer agent
- Computer restart
- Software Center
- Software deployment
- User and device affinity

For more information, see the following articles:

- [About client settings](#)
- [How to configure client settings](#)

Security permissions for application management

- The **Application Author** security role includes the required permissions to create, change, and retire applications.
- The **Application Deployment Manager** security role includes required permissions to deploy applications.
- The **Application Administrator** security role has all the permissions from both the **Application Author** and the **Application Deployment Manager** security roles.

For more information, see [Configure role-based administration](#).

App-V 4.6 SP1 or later client to run virtual applications

To create virtual applications in Configuration Manager, install App-V 4.6 SP1 or later on devices.

App-V is included with all supported versions of Windows 10 Enterprise edition. For more information, see [Getting started with App-V for Windows 10](#).

Remove the application catalog

Support ended for the application catalog roles with version 1910. Software Center can deliver all app deployments without the application catalog. For more information, see [Removed and deprecated features](#).

Starting in version 2107, you can't update the site if it has either of the application catalog site system roles. Remove these roles before you update to version 2107.

If your site still has an application catalog, use the following process to remove it:

1. Update all Configuration Manager clients to the latest supported version.
2. Set branding for Software Center, instead of in the properties of the application catalog web site role. For more information, see [Software Center client settings](#).
3. Review the default and any custom client settings. In the **Computer Agent** group, make sure the **Default Application Catalog website point** is **(none)**.
4. Remove the **application catalog website** and **application catalog web service** site system roles from all primary sites. For more information, see [Uninstall a site system role](#).

After you remove the application catalog roles, Software Center starts using the management point for user-targeted, available deployments. To verify this behavior on a specific client, review the `scclient_<username>.log`, and look for an entry similar to the following line:

```
Using endpoint Url: https://mp.contoso.com/CMUserService_WindowsAuth, Windows authentication
```

 **Note**

If you have any tools or automation that used the `ApplicationViewService.asmx` SOAP endpoint on the application catalog website point, you need to change it. Update the URL in your tool to use the management point user service endpoint. For example, `https://mp.contoso.com/CMUserService_WindowsAuth`

Next steps

[Plan for Software Center](#)

[Understand user notifications](#)

[Security and privacy for application management](#)

Plan for Software Center

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Users change settings, browse for applications, and install applications from Software Center. When you install the Configuration Manager client on a Windows device, it automatically installs Software Center as well.

Configure Software Center

Important

To take advantage of new Configuration Manager features, first update clients to the latest version. While new functionality appears in the Configuration Manager console when you update the site and console, the complete scenario isn't functional until the client version is also the latest.

Use client settings to configure the appearance and behaviors of Software Center. For more information, see [Software Center client settings](#). The following list is a summary of some of the configurations:

- Change the branding of Software Center to include your organization's name, colors, and logo. For more information, see [Brand Software Center](#).
- Configure which default tabs are visible, and add up to five custom tabs to Software Center.

In Configuration Manager 2103 and earlier, when single sign on with multifactor authentication is used, you may not be able to sign into custom tabs that load a website that's subject to conditional access policies.

- You can configure co-managed devices to use the Company Portal for both Intune and Configuration Manager apps. For more information, see [Use the Company Portal app on co-managed devices](#).

You can allow users to set in Software Center if they regularly use the computer for work. This option configures an affinity between the user and device, which can affect some deployments. For more information, see [Link users and devices with user device affinity](#).

Be aware of the following settings for features that are no longer supported:

- The client setting **Use new Software Center** in the **Computer Agent** group is enabled by default. The previous version of Software Center is no longer supported.
- The client setting **Hide application catalog link in Software Center** in the **Software Center Customizations** is enabled by default. This link would appear on the **Installation Status** tab of Software Center. The application catalog is no longer supported.

For more information, see [Removed and deprecated features](#).

Software Center and user-available applications

When you deploy an app with the purpose **Available** to a user collection, users can see these available applications in Software Center. This behavior provides a self-service capability for users to easily install approved software, without requiring assistance from IT staff.

Software Center gets application deployment information in policy from the management point. It uses the same management point from the assigned primary site as the Configuration Manager client. In a large environment, you can scale client communication to management points by assigning them to [boundary groups](#).

Users can browse and install user-available applications on Azure Active Directory (Azure AD)-joined devices and internet-based, domain-joined devices. For more information, see [Prerequisites to deploy user-available applications](#).

The site optimizes user-available deployments to reduce policy traffic between the server and clients. This behavior allows a large number of applications to be available for the user without significantly affecting performance of the overall infrastructure.

Support for enhanced HTTP

Starting in version 2107, Software Center can take advantage of enhanced HTTP when the management point is configured for HTTP. This site configuration provides secure communication without the overhead of managing PKI certificates. When you enable the site for enhanced HTTP, Software Center prefers secure communication over HTTPS to get user-available applications from the management point.



Tip

On any version of Configuration Manager, when you configure the site or the management point to require HTTPS communication, Software Center always uses HTTPS.

To validate this behavior, on a client review the following log files:

- **CCMSDKProvider.log**: Shows the client's selection of the HTTPS endpoint on the management point. For example: Management URL retrieved: https://...
- **SCClient_*.log**: Shows the endpoint URL that the client uses to communicate with the management point, which should use HTTPS. For example: Using endpoint Url: https://mp01.contoso.com:443/CMUserService, AAD authentication

Note

To take full advantage of new Configuration Manager features, after you update the site, also update clients to the latest version. The complete scenario isn't functional until the client version is also the latest.

For more information on how to configure the site, see [enhanced HTTP](#).

Brand Software Center

Change the appearance of Software Center to meet your organization's branding requirements. This configuration helps users trust Software Center.

Configure Software Center branding

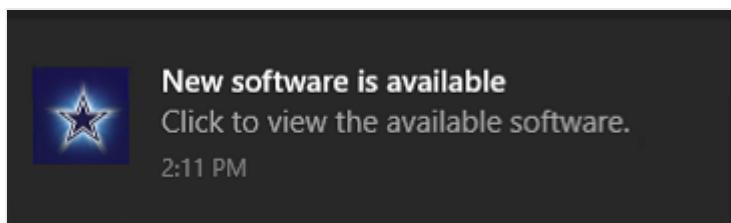
Customize the appearance of Software Center by adding your organization's branding elements:

- **Organization name**: Software Center displays this name in the top banner.
- **Color scheme**: The primary color for the banner and other elements.
- **Foreground color**: By default, when you select an item, the font color is white. Starting in version 2103, you can change this color for better visibility with certain primary colors, and better accessibility.
- **Logo for Software Center**: Your organization's logo helps users to trust Software Center.

The following image shows an example of Software Center that's customized with all four branding settings:

The screenshot shows the Software Center interface. At the top, there's a purple header bar with the 'Software Center' logo and the company name 'Test Company'. Below the header is a navigation sidebar on the left with icons for Applications (1 update), Updates, Operating Systems, Installation status, Device compliance, and Options. The main area displays a list of updates. The first item is titled 'Test MSI #2' and has a small image icon with a blue 'New' ribbon. Above the list are filters for 'All' and 'Required' and sorting options for 'Filter: All' and 'Sort by: Most recent'.

Starting in version 2111, you can also configure a **Logo for notifications**. It's a separate image file specifically for notifications on devices running Windows 10 or later. Your organization's logo helps users to trust these notifications. When you deploy software to a client, the user sees notifications with your logo. For example:



For more information, see the following articles:

- [About client settings for Software Center](#)
- [How to configure client settings](#)

Branding priorities

Configuration Manager applies the organization name for Software Center according to the following priorities:

1. **Software Center Customization** client setting for **Company Name**. For more information, see [About client settings: Software Center](#).
2. **Computer Agent** client setting for **Organization name**. For more information, see [About client settings: Computer agent](#).

Next steps

- [Software Center user guide](#)
- [Plan for and configure application management](#)
- [Use the Company Portal app on co-managed devices](#)

 **Note**

This article used to include more sections, which have moved to the following articles:

- [User notifications for required deployments](#)

User notifications

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

The Configuration Manager client and Software Center can display notifications to users that are signed-in to Windows. You can control many of these behaviors through client settings and the deployment settings.

ⓘ Note

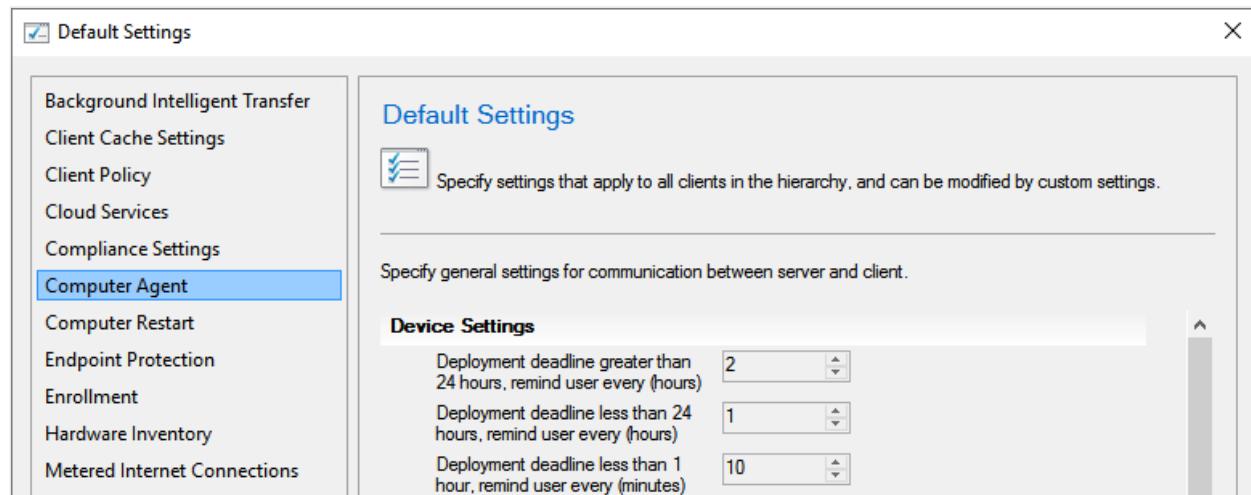
By default, Windows 11 enables **focus assist** for the first hour after a user signs on for the first time. For more information, see [Reaching the Desktop and the Quiet Period](#).

Software Center notifications are currently suppressed during this time. For more information, see [Turn Focus assist on or off in Windows ↗](#).

Required deployments

When users receive required software, and select the **Snooze and remind me** setting, they can choose from the following options:

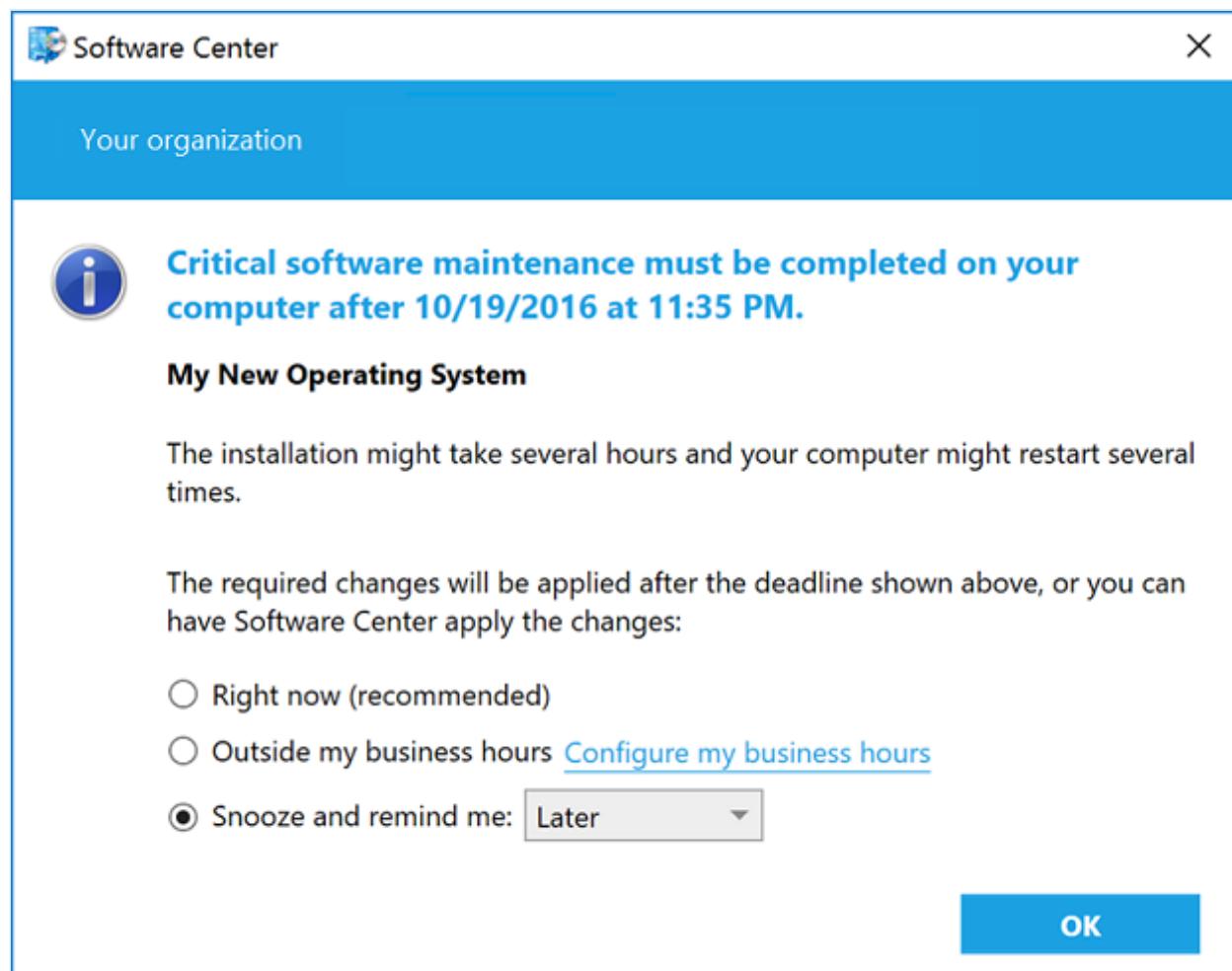
- **Later:** Specifies that notifications are scheduled based on the notification settings configured in client settings.
- **Fixed time:** Specifies that the notification is scheduled to display again after the selected time. For example, if you select 30 minutes, the notification displays again in 30 minutes.



The maximum snooze time is always based on the notification values configured in the client settings at every time along the deployment timeline. For example:

- You configure the **Deployment deadline greater than 24 hours, remind users every (hours)** setting on the Computer Agent page for 10 hours.
- The client displays the notification dialog more than 24 hours before the deployment deadline.
- The dialog shows snooze options up to but never greater than 10 hours.
- As the deployment deadline approaches, the dialog shows fewer options. These options are consistent with the relevant client settings for each component of the deployment timeline.

For a high-risk deployment, such as a task sequence that deploys an OS, the user notification experience is more intrusive. Instead of a transient taskbar notification, a dialog box like the following displays each time you're notified that critical software maintenance is required:



Replace toast notifications with dialog window

Sometimes users don't see the Windows toast notification about a restart or required deployment. Then they don't see the experience to snooze the reminder. This behavior can lead to a poor user experience when the client reaches a deadline.

When [software changes are required](#) or deployments [need a restart](#), you have the option of using a more intrusive dialog window.

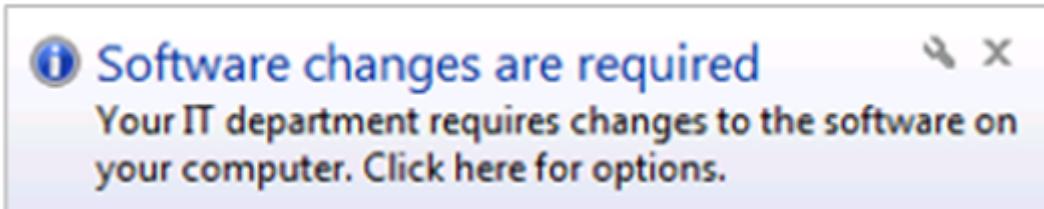
Software changes are required

When you [deploy an application](#) as required with a deadline in the future, on the **User Experience** page of the Deploy Software Wizard, select the following user notification options:

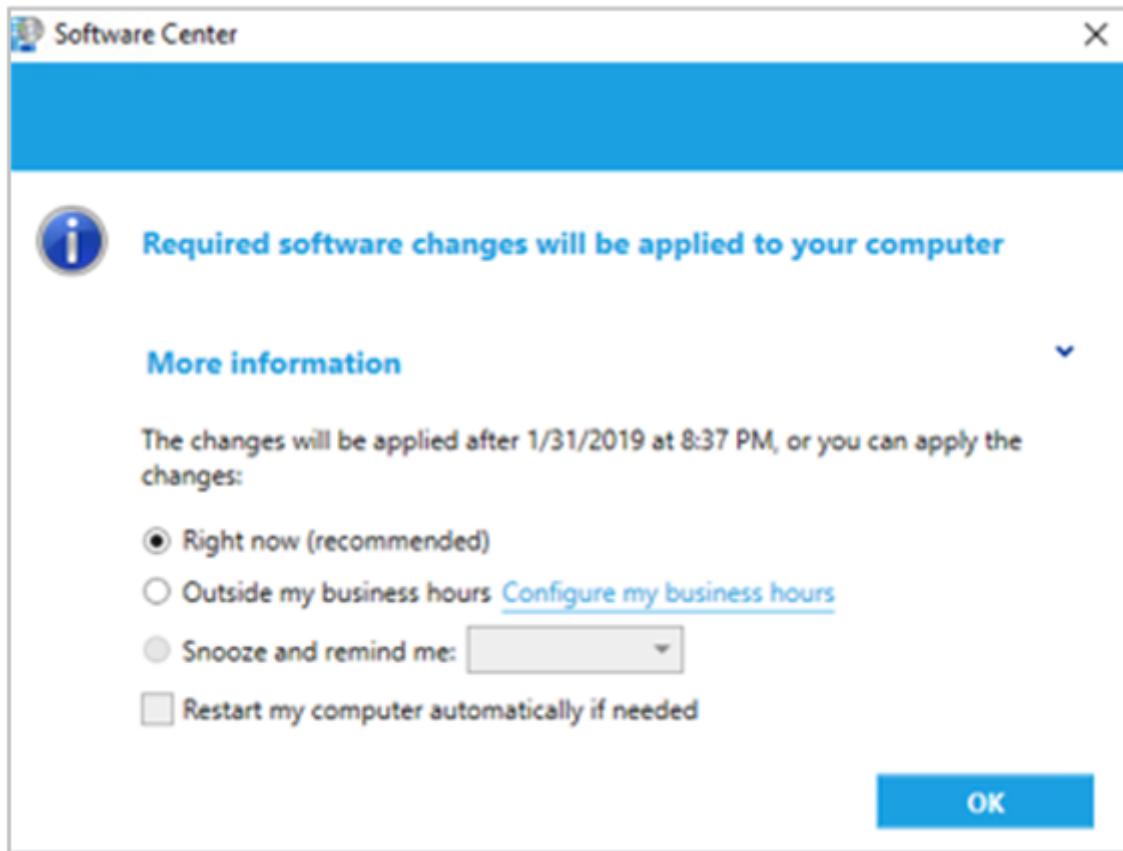
- Display in Software Center and show all notifications
- When software changes are required, show a dialog window to the user instead of a toast notification

Configuring this deployment setting changes the user experience for this scenario.

From the following toast notification:



To the following dialog window:



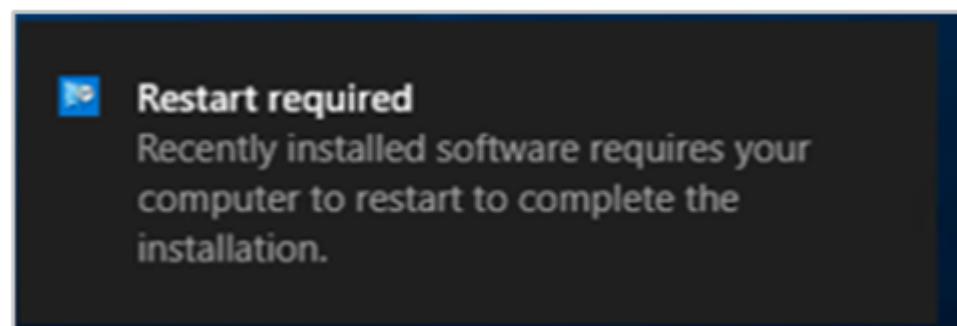
Restart required

In the [Computer Restart](#) group of client settings, enable the following option: **When a deployment requires a restart, show a dialog window to the user instead of a toast notification.**

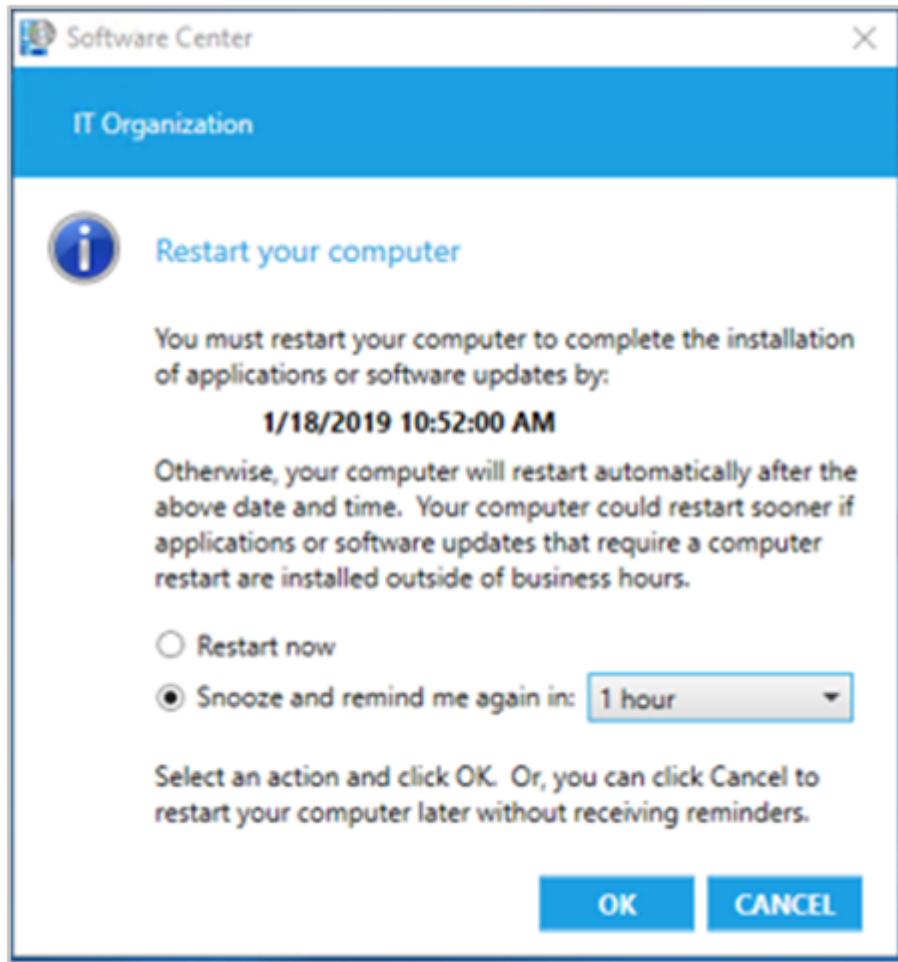
Configuring this client setting changes the user experience for all required deployments that require a restart of the following types:

- [Application](#)
- [Task sequence](#)
- [Software update](#)

From the following toast notification:



To the following dialog window:



Next steps

[Device restart notifications](#)

Security and privacy for application management in Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Security guidance

Centrally specify user device affinity

Manually specify the user device affinity instead of letting users identify their primary device. Don't enable usage-based configuration.

Don't consider information that's collected from users or from the device to be authoritative. If you deploy software by using user device affinity that a trusted administrator doesn't specify, the software might be installed on computers and to users who aren't authorized to receive that software.

Don't run deployments from distribution points

Always configure deployments to download content from distribution points rather than run from distribution points. When you configure deployments to download content from a distribution point and run locally, the Configuration Manager client verifies the package hash after it downloads the content. The client discards the package if the hash doesn't match the hash in the policy.

If you configure the deployment to run directly from a distribution point, the Configuration Manager client doesn't verify the package hash. This behavior means that the Configuration Manager client can install software that's been tampered with.

If you must run deployments directly from distribution points, use NTFS least permissions on the packages on the distribution points. Also use internet protocol security (IPsec) to secure the channel between the client and the distribution points, and between the distribution points and the site server.

Don't let users interact with elevated processes

If you enable the options to **Run with administrative rights** or **Install for system**, don't let users interact with those applications. When you configure an application, you can

set the option to **Allow users to view and interact with the program installation**. This setting allows users to respond to any required prompts in the user interface. If you also configure the application to **Run with administrative rights** or **Install for system**, an attacker at the computer that runs the program could use the user interface to escalate privileges on the client computer.

Use programs that use Windows Installer for setup and per-user elevated privileges for software deployments that require administrative credentials. Setup must be run in the context of a user who doesn't have administrative credentials. Windows Installer per-user elevated privileges provide the most secure way to deploy applications that have this requirement.

Note

When the user starts the application installation process from Software Center, the option to **Allow users to view and interact with the program installation** can't control user interactions with any other processes created by the application installer. Because of this behavior, even if you don't select this option, the user may still be able to interact with an elevated process. To avoid this issue, don't deploy applications that create other processes with user interactions. If you have to install this type of application, deploy it as **Required** and configure the user notification experience to **Hide in Software Center and all notifications**.

Restrict whether users can install software interactively

Configure the **Install permissions** client setting in the **Computer Agent** group. This setting restricts the types of users who can install software in Software Center.

For example, create a custom client setting with **Install permissions** set to **Only administrators**. Apply this client setting to a collection of servers. This configuration prevents users without administrative permissions from installing software on those servers.

For more information, see [About client settings](#).

For mobile devices, only deploy signed applications

Deploy mobile device applications only if they're code-signed by a certification authority (CA) that the mobile device trusts.

For example:

- An application from a vendor, which is signed by a public and globally trusted certificate provider.
- An internal application that you sign independent from Configuration Manager by using your internal CA.
- An internal application that you sign by using Configuration Manager when you create the application type and use a signing certificate.

Secure the location of the mobile device application signing certificate

If you sign mobile device applications by using the **Create Application Wizard** in Configuration Manager, secure the location of the signing certificate file, and secure the communication channel. To help protect against elevation of privileges and man-in-the-middle attacks, store the signing certificate file in a secured folder.

Use IPsec between the following computers:

- The computer that runs the Configuration Manager console
- The computer that stores the certificate signing file
- The computer that stores the application source files

Instead, sign the application independent of Configuration Manager and before you run the **Create Application Wizard**.

Implement access controls

To protect reference computers, implement access controls. When you configure the detection method in a deployment type by browsing to a reference computer, make sure that the computer isn't compromised.

Restrict and monitor administrative users

Restrict and monitor the administrative users who you grant the following application management role-based security roles:

- **Application Administrator**
- **Application Author**
- **Application Deployment Manager**

Even when you [configure role-based administration](#), administrative users who create and deploy applications might have more permissions than you realize. For example,

administrative users who create or change an application can select dependent applications that aren't in their security scope.

Configure App-V apps in virtual environments with the same trust level

When you configure Microsoft Application Virtualization (App-V) virtual environments, select applications that have the same trust level in the virtual environment. Because applications in an App-V virtual environment can share resources, like the clipboard, configure the virtual environment so that the selected applications have the same trust level.

For more information, see [Create App-V virtual environments](#).

Make sure macOS apps are from a trustworthy source

If you deploy applications for macOS devices, make sure that the source files are from a trustworthy source. The CMAppUtil tool doesn't validate the signature of the source package. Make sure the package comes from a source that you trust. The CMAppUtil tool can't detect whether the files have been tampered with.

Secure the cmmac file for macOS apps

If you deploy applications for macOS computers, secure the location of the `.cmmac` file. The CMAppUtil tool generates this file, and then you import it to Configuration Manager. This file isn't signed or validated.

Secure the communication channel when you import this file to Configuration Manager. To help prevent tampering with this file, store it in a secured folder. Use IPsec between the following computers:

- The computer that runs the Configuration Manager console
- The computer that stores the `.cmmac` file

Use HTTPS for web applications

If you configure a web application deployment type, use HTTPS to secure the connection. If you deploy a web application by using an HTTP link rather than an HTTPS link, the device could be redirected to a rogue server. Data that's transferred between the device and server could be tampered with.

Security issues

- Low-rights users can change files that record software deployment history on the client computer.

Because the application history information isn't protected, a user can change files that report whether an application is installed.

- App-V packages aren't signed.

App-V packages in Configuration Manager don't support signing. Digital signatures verify the content is from a trusted source and wasn't altered in transit. There's no mitigation for this security issue. Follow the security best practice to download the content from a trusted source and from a secure location.

- Published App-V applications can be installed by all users on the computer.

When an App-V application is published on a computer, all users who sign in to that computer can install the application. You can't restrict the users who can install the application after it's published.

Privacy information

Application management lets you run any application, program, or script on any client in the hierarchy. Configuration Manager has no control over the types of applications, programs, or scripts that you run or the type of information that they transmit. During the application deployment process, Configuration Manager might transmit information that identifies the device and sign-in accounts between clients and servers.

Configuration Manager maintains status information about the software deployment process. Unless the client communicates by using HTTPS, software deployment status information isn't encrypted during transmission. The status information isn't stored in encrypted form in the database.

The use of Configuration Manager application installation to remotely, interactively, or silently install software on clients might be subject to software license terms for that software. This use is separate from the Software License Terms for Configuration Manager. Always review and agree to the Software Licensing Terms before you deploy software by using Configuration Manager.

Configuration Manager collects diagnostics and usage data about applications, which is used by Microsoft to improve future releases. For more information, see [Diagnostics and usage data](#).

Application deployment doesn't happen by default and requires several configuration steps.

The following features help efficient software deployment:

- **User device affinity** maps a user to devices. A Configuration Manager administrator deploys software to a user. The client automatically installs the software on one or more computers that the user uses most often.
- **Software Center** is installed automatically on a device when you install the Configuration Manager client. Users change settings, browse for software, and install software from Software Center.

User device affinity privacy information

- Configuration Manager might transmit information between clients and management point site systems. The information might identify the computer, the sign-in account, and the summarized usage for sign-in accounts.
- Unless you configure the management point to require HTTPS communication, the information that's transmitted between the client and server isn't encrypted.
- The computer and sign-in account usage information is used to map a user to a device. Configuration Manager stores this information on client computers, sends it to management points, and then stores it in the site database. By default, the site deletes old information from the database after 90 days. The deletion behavior is configurable by setting the [Delete Aged User Device Affinity Data](#) site maintenance task.
- Configuration Manager maintains status information about user device affinity. Unless you [configure clients](#) to communicate with management points by using HTTPS, they don't encrypt status information during transmission. The site doesn't store status information in encrypted form in the database.
- Computer and sign-in usage information that's used to establish user and device affinity is always enabled. Users and administrative users can supply user device affinity information.

Software Center privacy information

- Software Center lets the Configuration Manager administrator publish any application, program, or script for users to run. Configuration Manager has no

control over the types of programs or scripts that are published in Software Center or the type of information that they transmit.

- Configuration Manager might transmit information between clients and the management point. The information might identify the computer and sign-in accounts. Unless you configure the management point to require clients connect by using HTTPS, the information that's transmitted between the client and servers isn't encrypted.
- The information about the application approval request is stored in the Configuration Manager database. For requests that are canceled or denied, the corresponding request history entries are deleted after 30 days by default. You can configure this deletion behavior with the [Delete Aged Application Request Data](#) site maintenance task. The site never deletes application approval requests that are in approved and pending states.
- When you install the Configuration Manager client on a device, it automatically installs Software Center.

Prerequisites to deploy user-available apps

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

When you deploy applications as **Available** to user collections, then users can browse Software Center and install the apps they need.

For on-premises domain-joined clients, Software Center uses the user's domain credentials to get the list of available applications from the management point.

There are other requirements for clients that are internet-based, joined to Azure Active Directory (Azure AD), or both.

Azure AD-joined devices

If you deploy applications as available to users, they can browse and install them through Software Center on Azure AD devices. Configure the following prerequisites to enable this scenario:

- Enable HTTPS on the management point or enable Enhanced HTTP on the site.
- Integrate the site with [Azure AD for Cloud Management](#).
 - Configure [Azure AD User Discovery](#).
- Deploy an application as available to a collection of users from Azure AD.
- Enable the client setting **Use new Software Center** in the [Computer agent](#) group.
- The client OS must be Windows 10 or later, and joined to Azure AD. Either as purely cloud domain-joined, or hybrid Azure AD-joined.
- To support internet-based clients:
 - Deploy a [cloud management gateway](#) (CMG).
 - Distribute any application content to a content-enabled CMG.
 - Enable the client setting: **Enable user policy requests from Internet clients** in the [Client Policy](#) group.
- To support clients on the intranet:

- Add the content-enabled CMG to a boundary group used by the clients.
- Clients must resolve the fully qualified domain name (FQDN) of the management point.

① Note

For a client detected as on the intranet, but communicating via the cloud management gateway (CMG), it uses Azure Active Directory (Azure AD) identity for devices joined to Azure AD. These devices can be cloud-joined or hybrid-joined.

Internet-based domain-joined devices

An internet-based, domain-joined device that isn't joined to Azure AD and communicates via a cloud management gateway (CMG) can get apps deployed as available. The Active Directory domain user of the device needs a matching Azure AD identity. When the user starts Software Center, Windows prompts them to enter their Azure AD credentials. They can then see any available apps.

Configure the following prerequisites to enable this functionality:

- Windows 10 or later device, and:
 - Joined to your on-premises Active Directory domain.
 - Can communicate via [CMG](#).
- The site has discovered the user by *both* [Active Directory](#) and [Azure AD user discovery](#).

① Note

If you apply a **software restriction policy** to the device, it can block the authentication prompt in Windows. Review any domain or local group policies that you apply to the device. Then remove any that might interfere with this Software Center behavior.

Next steps

[Deploy applications](#)

Create applications in Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

A Configuration Manager application defines the metadata about application. An application has one or more deployment types. These deployment types include the installation files and information that are required to install software on devices. A deployment type also has rules, such as detection methods, and requirements. These rules specify when and how the client installs the software.

Create applications using the following methods:

- Automatically create an application and deployment types by reading the application installation files:
 - [Create an application](#) and [automatically detect](#) application information
 - [Create a deployment type](#) and [automatically identify](#) deployment type information
- Manually create an application and then add deployment types later:
 - [Create an application](#) and [manually specify](#) application information
 - [Create a deployment type](#) and [manually specify](#) deployment type information
- [Import an application](#) from a file

This article also includes the following information to configure a deployment type:

- [Content](#)
- [Task Sequence](#)
- [Detection Method](#)
- [User Experience](#)
- [Requirements](#)
- [Return Codes](#)
- [Dependencies](#)

Create an application

1. In the Configuration Manager console, go to the **Software Library** workspace, expand **Application Management**, and select the **Applications** node.

2. On the **Home** tab of the ribbon, in the **Create** group, select **Create Application**.

Next, automatically detect or manually specify application information:

- **Automatically detect** application information to create a basic application with a single deployment type. For example, a Windows Installer file that has no dependencies or requirements. After you create an application by using this procedure, edit it as needed. You can add or change deployment types, and add detection methods, dependencies, or requirements.
- **Manually specify** application information to create more complex applications. Define more than one deployment type, dependencies, detection methods, or requirements.

Automatically detect application information

1. On the **General** page of the Create Application wizard, select **Automatically detect information about this application from installation files**.
2. In the **Type** drop-down list, select the application installation file type that you want to use to detect application information. For more information about the available installation types, see [Deployment types supported by Configuration Manager](#).
3. In the **Location** box, specify the application installation file that you want to use to detect application information. This location is either a network path (`\server\share\filename`) or a store link. You must have access to the network path and any subfolders that include application content.

Important

When you select **Windows Installer (*.msi file)** as an application type, the site imports all of the files in the specified folder. It then sends these files to distribution points. Make sure that the specified folder contains only the files that are necessary to install the application. Microsoft tests Configuration Manager to support up to 20,000 files in the application package. If your application has more files, consider creating multiple applications with less files.

4. On the **Import Information** page of the Create Application wizard, review the information, and then select **Next**. If necessary, select **Previous** to go back and fix any errors.

5. On the **General Information** page of the Create Application wizard, specify the following information:

 **Note**

If Configuration Manager automatically detects this information from the application installation files, it's already populated here. Additionally, the displayed options might be different depending on the application type that you create.

- General information about the application, like the application **Name**, **Administrator comments**, **Publisher**, and **Software version**. To help you find the application in the Configuration Manager console, specify an **Optional reference**, or select **Administrative categories**.
- **Installation program**: Specify the installation program and any required properties that are needed to install the application deployment type.

 **Tip**

If the installation program doesn't appear, choose **Browse** and browse to the installation program location.

- **Install behavior**: Select one of the three options for how Configuration Manager installs this deployment type. For more information on these options, see [User Experience](#).
- **Use an automatic VPN connection (if configured)**: If you've deployed a VPN profile to the device on which the user launches the app, connect the VPN when the app starts. This option is only for Windows 8.1 and Windows Phone 8.1. On Windows Phone 8.1 devices, if you deploy more than one VPN profile to the device, automatic VPN connections aren't supported. For more information, see [VPN profiles](#).
- **Provision this application for all users on the device**: Provision an application with a Windows app package for all users on the device. For more information, see [Create Windows applications](#).

 **Tip**

If you're modifying an existing application, this setting is on the **User Experience** tab of the Windows app package deployment type properties.

6. Choose **Next**, review the application information on the **Summary** page, and then finish the Create Application wizard.

The new application now appears in the **Applications** node of the Configuration Manager console. You've finished creating an application.

To add more deployment types or configure other settings, see [Create deployment types for the application](#).

Manually specify application information

1. On the **General** page of the Create Application wizard, select **Manually specify the application information**, and then choose **Next**.
2. Specify **General Information** about the application:
 - The application **Name** is required and must be fewer than 256 characters.
 - **Administrator comments**, **Publisher**, and **Software version** are additional metadata to further describe the application.
 - To help you find the application in the Configuration Manager console, specify an **Optional reference**, or select **Administrative categories**.
 - **Date published**
 - Select users or groups who are responsible for this application as **Owners** and **Support contacts**. By default, these values are set to your username.
3. On the **Software Center** page of the Create Application wizard, specify the following information:
 - **Selected language**: In the drop-down list, select the language version of the application that you want to set up. Choose **Add/Remove** to set up more languages for this application.
 - **Localized application name**: Specify the application name in the selected language.

 **Important**

A localized application name is required for each language version that you set up.

- **User categories:** Choose **Edit** to specify application categories in the selected language. Users of Software Center use these categories to help filter and sort the applications.

 **Note**

User categories for device-targeted application deployments show as filters in Software Center. These deployments can be either available or required.

Renaming or deleting a category doesn't automatically apply to apps with this category. These changes apply on the next revision of the app.

To work around this issue for rename or delete:

- First clear the checkbox for the category on any app that references it. Then apply that change, which revises the app.
- Instead of the rename action, next create a new category with the new name, and add the new category to the relevant apps.
- You can delete the category after you revise the apps.

- **User documentation:** Specify the location of a file from which Software Center users can get more information about this application. This location is a website address, or a network path and file name. Make sure that users have access to this location.
- **Link text:** Specify the text that appears in place of "Additional information" when user documentation is specified.
- **Privacy URL:** Specify a website address to the privacy statement for the application.
- **Localized description:** Enter a description for this application in the selected language.
- **Keywords:** Enter a list of keywords in the selected language. These keywords help Software Center users search for the application.
- **Icon:** Select **Browse** to select an icon for this application. If you don't specify an icon, Configuration Manager uses a default icon. Icons can have pixel dimensions of up to 512x512.

4. On the **Deployment Types** page of the Create Application wizard, choose **Add** to create a new deployment type. For more information, see [Create deployment types for the application](#).
5. Choose **Next**, review the application information on the **Summary** page, and then finish the Create Application wizard.

The new application now appears in the **Applications** node of the Configuration Manager console.

Create deployment types for the application

If you [automatically detect application information](#), you may not need to finish some of the steps in this section.

Note

When you view the properties of an existing deployment type, the following sections correspond to tabs of the deployment type properties window:

- Content
- Task Sequence
- Detection Method
- User Experience
- Requirements
- Return Codes
- Dependencies

For information on the **Install Behavior** tab on the properties of a deployment type, see [Check for running executable files](#).

Start the Create Deployment Type wizard

There are three ways to start the Create Deployment Type wizard:

- **In the Applications node:** In the Configuration Manager console, go to the **Software Library** workspace, expand **Application Management**, and select the **Applications** node. Select an application, and then select **Create Deployment Type** in the ribbon.

- **When creating an application:** When you [Manually specify application information](#) in the Create Application wizard, select **Add** on the Deployment Types page.
- **From application properties:** Select an existing application in the **Applications** node and select **Properties**. Switch to the **Deployment Types** tab, and select **Add**.

Then use one of the following procedures to [automatically identify](#) or [manually specify](#) deployment type information.

Automatically identify deployment type information

1. On the **General** page of the Create Deployment Type wizard:
 - a. Select the application installation file **Type** to detect the deployment type information.
 - b. Select **Automatically identify information about this deployment type from installation files**.
 - c. In the **Location** box, specify the application installation file that you want to use to detect the deployment type information. This location is either a network path (`\server\share\filename`) or a store link. You must have access to the network path and any subfolders that include application content.
2. On the **Import Information** page of the Create Deployment Type wizard, review the information, and then select **Next**. If necessary, select **Previous** to go back and fix any errors.
3. On the **General Information** page of the Create Deployment Type wizard, specify the following information:

Note

Some of the deployment type information might already be present if it was read from the application installation files. Additionally, the displayed options might differ, depending on the deployment type that you're creating.

- **General Information** about the deployment type:
 - The **Name** is required
 - **Administrator comments** to further describe it
 - **Languages** that are available for it

- **Installation program:** Specify the installation program and any properties that you require to install the deployment type.
- **Install behavior:** Select one of the three options for how Configuration Manager installs this deployment type. For more information on these options, see [User Experience](#).
- **Use an automatic VPN connection (if configured):** If you've deployed a VPN profile to the device on which the user launches the app, connect the VPN when the app starts. This option is only for Windows 8.1 and Windows Phone 8.1. On Windows Phone 8.1 devices, if you deploy more than one VPN profile to the device, automatic VPN connections aren't supported. For more information, see [VPN profiles](#).

4. Choose **Next**, and then continue to [Deployment type Content options](#).

Manually specify the deployment type information

1. On the **General** page of the Create Deployment Type wizard, in the **Type** drop-down list, choose the application installation file type for this deployment type.
2. Select **Manually specify the deployment type information**, and then select **Next**.
3. On the **General Information** page of the Create Deployment Type wizard, specify a **Name** for the deployment type. Optionally specify **Administrator comments**, select the **Languages** for this deployment type, and then select **Next**.
4. Continue to [Deployment type Content options](#).

Deployment type Content options

On the **Content** page, specify the following information:

Note

When you view the properties of an existing deployment type, some of these options appear on the **Content** tab and some on the **Programs** tab.

- **Content location:** Specify the location of the content for this deployment type, or select **Browse** to choose the deployment type content folder.

Important

The System account of the site server computer must have permissions to the specified content location.

- **Persist content in the client cache:** The Configuration Manager client indefinitely keeps in its cache the deployment type content. The client persists the content even if the app is already installed. This option is useful with some deployments, like Windows Installer-based software. Windows Installer needs a local copy of the source content for applying updates. This option reduces the available cache space. If you select this option, it might cause a large deployment to fail at a later point if the cache doesn't have sufficient available space.

Tip

This option persists the specific version of content that the client installs. If you update the content for this app, the client doesn't automatically cache this content again. Once an action happens that requires the new content, the client downloads the new content version.

- **Installation program:** Specify the name of the installation program and any required installation parameters.
 - **Installation start in:** Optionally specify the folder that has the installation program for the deployment type. This folder can be an absolute path on the client or a path to the distribution point folder that has the installation files.
- **Uninstall program:** Optionally specify the name of the uninstall program and any required parameters.
 - **Uninstall start in:** Optionally specify the folder that has the uninstall program for the deployment type. This folder can be an absolute path on the client. It can also be a relative path on a distribution point of the folder with the package.
- **Repair program:** For Windows Installer and Script Installer deployment types, optionally specify the name of the repair program and any required parameters.
 - **Repair start in:** Optionally specify the folder that has the repair program for the deployment type. This folder can be an absolute path on the client. It can also be a relative path on a distribution point of the folder with the package.
- **Run installation and uninstall program as 32-bit process on 64-bit clients:** Use the 32-bit file and registry locations on Windows-based computers to run the installation program for the deployment type.

Deployment type properties Content options

When you view the properties of a deployment type, the following options appear only on the **Content** tab:

- **Uninstall content settings:**
 - **Same as install content:** If the install and uninstall content are the same, select this option. This option is the default.
 - **No uninstall content:** If your application doesn't need content for uninstall, select this option.
 - **Different from install content:** If the uninstall content is different from the install content, select this option.
 - **Uninstall content location:** Specify the network path to the content that's used to uninstall the application.
- **Allow clients to use distribution points from the default site boundary group:** Specify if clients should download and install the software from a distribution point in the site default boundary group when the content isn't available from a distribution point in the current or neighbor boundary groups.
- **Deployment options:** Specify if clients should download the application when they use a distribution point from a neighbor or the default site boundary groups.

ⓘ Note

Windows BranchCache is always enabled on clients. If the distribution point supports BranchCache, clients use it. For more information, see [BranchCache](#).

Deployment type Task Sequence options

For more information on the task sequence deployment type, see [Task sequence deployment type](#).

On the **Task Sequence** page, specify the following information:

- **Install task sequence:** Select a task sequence that runs the installation process for this app.
- **Uninstall task sequence (optional):** Select a task sequence that removes this app.

💡 Tip

If your task sequence doesn't appear in the list, double-check that it doesn't include any OS deployment or OS upgrade steps. Also confirm that it isn't marked as a high-impact task sequence. For more information, review the prerequisites for the [Task sequence deployment type](#).

Deployment type Detection Method options

This procedure sets up a detection method that indicates the presence of the deployment type. In other words, whether the Windows device already has the application installed. Use one of the two following methods to create a detection method:

- [Configure rules to detect the presence of this deployment type](#)
- [Use a custom script to detect the presence of this deployment type](#)

Configure rules to detect the presence of this deployment type

1. On the **Detection Method** page, the option to [Configure rules to detect the presence of this deployment type](#) is selected by default. Select **Add Clause**.
2. In the **Detection Rule** dialog box, select a **Setting type** to detect the presence of the deployment type:
 - **File System:** Detect whether a specified file or folder exists on a device. This detection indicates that the application is installed. Specify the following additional details:
 - **Type:** Select whether it's a file or folder.
 - **Path (Required):** Enter or browse to the local path on the device that includes the file or folder. For example, `C:\Program Files`. You can't specify a shared network path. If you select **Browse**, browse the local file system or connect to a representative client to browse.
 - **File or folder name (Required):** Specify the specific file or folder name to detect in the above path. If the client detects this file or folder on the device, it considers the application as installed on the device.
 - **This file or folder is associated with a 32-bit application on 64-bit systems:** The client first checks 32-bit file locations for the specified file or

folder. If the file or folder isn't found, the client then searches 64-bit locations.

- **Registry:** Detect whether a specified registry key or registry value exists on a client device. This detection indicates that the application is installed. Specify the following additional details:
 - **Hive** (Required): Choose a registry hive from the drop-down list. For example, `HKEY_LOCAL_MACHINE`.
 - **Key** (Required): Specify the registry key to search in the above hive. For example, `SOFTWARE\Microsoft\Office`.
 - **Value** (Optional): Enter a specific value to detect in the above key. If you want the client to detect the (Default) value, enable the option to **Use (Default) registry key value for detection**. When you enter a value or enable this option, you're required to select a **Data Type**.
 - **This registry key is associated with a 32-bit application on 64-bit systems:** Select this option to first check 32-bit registry locations for the specified registry key. If the registry key isn't found, the client searches 64-bit locations.
- **Windows Installer:** Detect whether a specified Windows Installer file exists on a client device. This detection indicates that the application is installed. Specify the **MSI Product code** to detect on the client. If you select **Browse**, choose the MSI file from which to read the product code.

3. At the bottom of the Detection Rule window, specify whether the item must exist or satisfy a rule. For example, if you detect with a file, the following option is selected by default: **The file system setting must exist on the target system to indicate presence of this application**. Select the other option to create a rule for detection based on file or folder properties. These properties include Date Modified, Date Created, Version, or Size. These rule criteria are different for each setting type.

4. Select **OK** to close the **Detection Rule** dialog box.

When you create more than one detection method for a deployment type, you can group clauses together to create more complex logic.

Group detection clauses (*optional*)

1. Create three or more detection method clauses on a deployment type.

2. Select two or more consecutive clauses, and then select **Group**. You'll see the parentheses added to the associated columns, which show where the group starts and ends.

Example:

Connector	(Clause)
		MSI Product Code	
Or	(file1.text exists	
And		file2.txt exists)

3. To remove the group, select the grouped clauses, and then select **Ungroup**.

Continue to the next section on using a custom script as a detection method. Or *skip* to the [User Experience](#) options for the deployment type.

Use a custom script to check for the presence of a deployment type

1. On the **Detection Method** page, select the **Use a custom script to detect the presence of this deployment type** box. Then select **Edit**.
2. In the **Script Editor** dialog box, select a **Script type** to detect the deployment type: PowerShell, VBScript, or JScript.

① Note

When a Windows PowerShell script runs as a app detection method, the Configuration Manager client calls PowerShell with the `-NoProfile` parameter. This option starts PowerShell without profiles. A PowerShell profile is a script that runs when PowerShell starts.

3. In the **Script contents** box, enter the script that you want to use, or paste in the contents of an existing script. Choose **Open** to browse to an existing saved script. Select **Clear** to remove the text in the Script contents field. If necessary, enable the option to **Run script as 32-bit process on 64-bit clients**.

① Note

The maximum size for a script is 32 KB.

4. Select **OK** to save the script and close the **Script Editor** dialog box. Back on the **Create Deployment Type** wizard, the **Script Type** and **Script Length** fields update with details about your script.

About custom script detection methods

Configuration Manager checks the results from the script. It reads the values written by the script to the standard output (STDOUT) stream, the standard error (STDERR) stream, and the exit code. If the script exits with a non-zero value, the script fails, and the application detection status is *Unknown*. If the exit code is zero, and STDOUT has data, the application detection status is *Installed*.

Tip

When writing a detection script, if you return a zero exit code but don't return output (data in STDOUT), the application will not be detected as installed. For more information, see the following examples.

Use the following tables to check whether an application is installed from the output from a script:

Zero exit code

STDOUT	STDERR	Script result	Application detection state
Empty	Empty	Success	Not installed
Empty	Not empty	Failure	Unknown
Not empty	Empty	Success	Installed
Not empty	Not empty	Success	Installed

Non-zero exit code

STDOUT	STDERR	Script result	Application detection state
Empty	Empty	Failure	Unknown
Empty	Not empty	Failure	Unknown
Not empty	Empty	Failure	Unknown

STDOUT	STDERR	Script result	Application detection state
Not empty	Not empty	Failure	Unknown

Examples

Use the following PowerShell/VBScript examples to write your own application detection scripts:

Example 1: The script returns an exit code that's not zero. This code indicates the script failed to run successfully. In this case, the application detection state is unknown.

PowerShell

```
Exit 1
```

VBScript

```
WScript.Quit(1)
```

Example 2: The script returns an exit code of zero, but the value of STDERR isn't empty. This result indicates the script failed to run successfully. In this case, the application detection state is unknown.

PowerShell

```
Write-Error "Script failed"
Exit 0
```

VBScript

```
WScript.Stderr.Write "Script failed"
WScript.Quit(0)
```

Example 3: The script returns an exit code of zero, which indicates it ran successfully. However, the value for STDOUT is empty, which indicates the application isn't installed.

PowerShell

```
Exit 0
```

VBScript

```
WScript.Quit(0)
```

Example 4: The script returns an exit code of zero, which indicates it ran successfully. The value for STDOUT isn't empty, which indicates the application is installed.

PowerShell

```
Write-Host "The application is installed"  
Exit 0
```

VBScript

```
WScript.Stdout.Write "The application is installed"  
WScript.Quit(0)
```

Example 5: The script returns an exit code of zero, which indicates it ran successfully. The values for STDOUT and STDERR aren't empty, which indicates the application is installed.

PowerShell

```
Write-Host "The application is installed"  
Write-Error "Completed"  
Exit 0
```

VBScript

```
WScript.Stdout.Write "The application is installed"  
WScript.Stderr.Write "Completed"  
WScript.Quit(0)
```

Deployment type User Experience options

These settings specify how the client installs the application on devices, and what the user sees.

On the **User Experience** page, specify the following information:

- **Installation behavior:** In the drop-down list, select one of the following options:
 - **Install for user:** The client only installs the application for the user to whom you deploy the application.

- **Install for system:** The client installs the application only once. It's available to all users.
- **Install for system if resource is device; otherwise, install for user:** If you deploy the application to a device, the client installs it for all users. If you deploy the application to a user, the client only installs it for that user.
- **Logon requirement:** Select one of the following options:
 - Only when a user is logged on
 - Whether or not a user is logged on
 - Only when no user is logged on

 **Note**

This option defaults to **Only when a user is logged on**. If you select **Install for user** in the **Installation behavior** drop-down list, you can't change this option.

- **Installation program visibility:** Specify the mode in which the deployment type runs on client devices. Select one of the following options:
 - **Maximized:** The deployment type runs maximized on client devices. Users see all installation activity.
 - **Normal:** The deployment type runs in the normal mode based on system and program defaults. This mode is the default.
 - **Minimized:** The deployment type runs minimized on client devices. Users might see the installation activity in the notification area or taskbar.
 - **Hidden:** The deployment type runs hidden on client devices. Users see no installation activity.
- **Allow users to view and interact with the program installation:** Specify whether a user can interact with the deployment type installation to set up the installation options.

If you selected the **Install for user** option in the **Installation behavior** drop-down list, this option is enabled by default.

 **Important**

When you select the **Install for system** behavior, this setting is optional. This change is primarily to allow an end user to interact with the installation during a task sequence. For example, to run a setup process that prompts the end user for various options. Some application installers can't have user prompts silenced, or the installation process may require specific configuration values only known to the user.

Installing in system context and allowing users to interact with the installation isn't a secure configuration. For more information, see [security and privacy for application management](#).

- **Maximum allowed run time (minutes):** Specify the maximum time in minutes that you expect the deployment type to run on the client computer. Specify this setting as a whole number greater than zero. The default value is 120 minutes (two hours).

Use this value for the following actions:

- To monitor the results from the deployment type.
- To check whether a deployment type is installed when you define maintenance windows on client devices. When a maintenance window is in place, a deployment type only starts if enough time is available in the maintenance window to accommodate the **Maximum Allowed Run Time** setting.

 **Important**

A conflict might occur if the **Maximum allowed run time** is longer than the scheduled maintenance window. If the user sets the maximum run time to a period greater than the length of any available maintenance window, that deployment type doesn't run.

- **Estimated installation time (minutes):** Specify the estimated installation time of the deployment type. Users see this time in Software Center.

Deployment type properties User Experience options

When you view the properties of a deployment type, the following options appear only on the **User Experience** tab:

Enforce specific post-installation behavior. Select one of the following options:

- **Determine behavior based on return codes:** Handle reboots based on the codes configured on the [Return Codes](#) tab. Software Center displays **Might Require a Reboot**. If a user is signed in during the install, they're prompted depending on the *deployment's* User Experience configuration.
- **No specific action:** No reboot required after installation. Software Center reports that no reboot is required.
- **The software install program might force a device restart:** Configuration Manager doesn't control or initiate a reboot, but the actual installation might do so without warning. Use this setting to prevent Configuration Manager from reporting installation failure when the installer initiates a reboot. Software Center displays **Might Require a Reboot**.
- **Configuration Manager client will force a mandatory device restart:** Configuration Manager forces a device reboot after successful installation. Software Center reports that a reboot is required. If a user is signed in during the install, they're prompted depending on the *deployment's* User Experience configuration.

Deployment type Requirements

Configuration Manager verifies these requirements on devices before installing the deployment type. Use requirements to further refine and control the devices or users that receive this application. For example, if you deploy the application to a user collection, specify the app's hardware requirements here.

1. On the **Requirements** page, select **Add** to open the **Create Requirement** dialog box.
2. In the **Category** drop-down list, select whether this requirement is for a **Device** or a **User**.

Select **Custom** to use a previously created global condition. When you select **Custom**, you can also choose **Create** to create a new global condition. For more about global conditions, see [How to create global conditions](#).

Important

If you deploy the application to a device collection, the client ignores any requirement of the category **User** and the condition **Primary Device**.

3. In the **Condition** drop-down list, select the condition to assess whether the user or device meets the installation requirements. The contents of this list vary depending on the selected category.
4. In the **Operator** drop-down list, select the operator to use. This operator compares the selected condition to the specified value. It assesses whether the user or device meets the installation requirement. The available operators vary depending on the selected condition. When using the **One of** operator, the **Values** field has validation that you have to enter one entry per row.

 **Note**

The available requirements differ depending on the device type that the deployment type uses.

5. In the **Value** box, specify the values to use for comparison. These values, along with the selected condition and operator, evaluate whether the user or device meets the installation requirements. The available values vary depending on the selected condition and the selected operator.
6. Choose **OK** to save the requirement and close the **Create Requirement** dialog box.

Deployment type Dependencies

Dependencies define one or more deployment types from another application that the client must install before it installs this deployment type.

 **Important**

In some cases, a deployment type is dependent on a deployment type that also has dependencies. The maximum number of supported dependencies in the chain is five.

1. On the **Dependencies** page, select **Add**.
2. In the Add Dependency window, enter the **Dependency group name**. This name refers to this group of application dependencies.
3. In the Add Dependency window, select **Add**.
4. In the **Specify Required Application** window, select an available application and at least one of its deployment types to use as a dependency.

Tip

Select **View** to display the properties of the selected application or deployment type.

5. Select **OK** to close the **Specify Required Application** window.
6. If you want the client to automatically install the dependent application, select **Auto Install** next to the dependency.

Note

You don't need to deploy a dependent application for the client to automatically install it.

7. If you add more than one dependency, use the **Increase Priority** and **Decrease Priority** buttons. These actions change the order in which the client evaluates each dependency.
8. Select **OK** to close the **Add Dependency** window.

Deployment type Return Codes

Note

This page isn't in the Create Deployment Type wizard. It's only a tab on the properties of an existing deployment type.

Specify return codes to control behaviors after the deployment type completes. For example, signal that a restart is required, the installation is complete.

1. On the **Return Codes** tab of the deployment type properties window, select **Add**.
2. In the Add Return Code window, specify the **Return Code Value** that you expect from this deployment type. This value is any positive or negative integer between **-2147483648** and **2147483647**.
3. Select a **Code Type** from the drop-down list. This setting defines how Configuration Manager interprets the specified return code from this deployment type. The available types vary based on the deployment type technology.

- **Success (no reboot)**: The deployment type successfully installed, and no reboot is necessary.
- **Failure (no reboot)**: The deployment type failed to install.
- **Hard Reboot**: The deployment type successfully installed, but requires the device to restart. Nothing else can be installed until the device restarts.
- **Soft Reboot**: The deployment type successfully installed, but requests the device to restart. Other installations can occur before the device restarts.
- **Fast Retry**: Another installation is already in progress on the device. The client retries every two hours, for a total of 10 times.

4. Optionally, enter a **Name** and **Description** for this return code.

5. Select **OK** to close the Add Return Code window.

Example: non-zero success

You're deploying an application that returns an exit code of **1** when it successfully installs. By default, Configuration Manager detects this non-zero return code as a failure. Specify the Return Code Value of **1**, and select the Code Type of **Success (no reboot)**. Now Configuration Manager interprets that return code as a success for this deployment type.

Default return codes

When you create some deployment types, Configuration Manager automatically adds the following return codes that are common to that technology:

Windows Installer (*.msi file)

Value	Code Type
0	Success (no reboot)
1707	Success (no reboot)
3010	Soft Reboot
1641	Hard Reboot
1618	Fast Retry

Script Installer

Value	Code Type
0	Success (no reboot)
1641	Hard Reboot
3010	Soft Reboot
1618	Fast Retry

Windows app package (*.appx, *.appxbundle, *.msix, *.msixbundle)

Value	Code Type
15605	Fast Retry
15618	Fast Retry

Additional options for App-V deployment types

Configure additional options that are unique to deployment types for virtual applications (App-V).

App-V deployment type Content options

1. In the Configuration Manager console, go to the **Software Library** workspace, expand **Application Management**, and select the **Applications** node.
2. Select an application with an App-V deployment type, and select **Properties**.
3. In the application properties, switch to the **Deployment Types** tab. Select the App-V deployment type, and select **Edit**.
4. In the deployment type properties, switch to the **Content** tab. Configure the following options as necessary:
 - **Persist content in the client cache:** The Configuration Manager client won't delete from its cache the content for this deployment type.

- **Load content into App-V cache before launch:** Before the application starts, the Configuration Manager client loads into the App-V cache all content for this deployment type. The client doesn't pin the content in the cache. It deletes the content as necessary.

5. Select **OK** to close the deployment type properties. Then select **OK** to close the application properties.

App-V deployment type Publishing options

1. In the Configuration Manager console, go to the **Software Library** workspace, expand **Application Management**, and select the **Applications** node.
2. Select an application with an App-V deployment type, and select **Properties**.
3. In the application properties, switch to the **Deployment Types** tab. Select the App-V deployment type, and select **Edit**.
4. In the deployment type properties, switch to the **Publishing** tab. Select the items in the virtual application that you want to publish.
5. Select **OK** to close the deployment type properties. Then select **OK** to close the application properties.

Import an application

Use the following procedure to import an application into Configuration Manager:

1. In the Configuration Manager console, go to the **Software Library** workspace, expand **Application Management**, and select the **Applications** node.
2. In the ribbon, on the **Home** tab and the **Create** group, select **Import Application**.
3. On the **General** page of the Import Application Wizard, specify the network path to the **File** to import. For example, `\server\share\file.zip`. This file is a valid compressed archive (ZIP format) of an exported Configuration Manager application.
4. On the **File Content** page, select the action to take if this application is a duplicate of an existing application. Create a new application, or ignore the duplicate and add a new revision to the existing application.
5. On the **Summary** page, review the actions, and then finish the wizard.

The new application appears in the **Applications** node.

 **Tip**

The Windows PowerShell cmdlet **Import-CMApplication** has the same function as this procedure. For more information, see [Import-CMApplication](#).

For more information about how to export an application, see [Management tasks for applications](#).

Supported deployment types

Configuration Manager supports the following deployment types for applications:

Deployment type	Description
name	
Windows Installer (*.msi file)	A Windows Installer file (.msi).
Windows app package (*.appx, *.appxbundle, *.msix, *.msixbundle)	Windows app package files (.appx or .msix) or Windows app bundle packages (.appxbundle or .msixbundle).
Windows app package (in the Windows Store)	Specify a link to the app in the Windows Store, or browse the store to select the app. Note 1
Script Installer	Specify a script or program that runs on Windows clients to install content or to do an action. Use this deployment type for setup.exe installers or script wrappers.
Microsoft Application Virtualization 4	A Microsoft App-V v4 manifest.
Microsoft Application Virtualization 5	A Microsoft App-V v5 package file.
Windows Phone app package (*.xap file)	A Windows Phone app package file.

Deployment type name	Description
Windows Phone app package (in the Windows Phone Store)	Specify a link to the app in the Windows Store.
macOS X	For macOS computers running the Configuration Manager client. Create a <code>.cmmac</code> file with the CMApUtil tool.
Web Application	Specify a link to a web application. This deployment type installs a shortcut to the web application on the user's device.
Windows Installer through MDM (*.msi)	Create and deploy Windows Installer-based apps to Windows devices using on-premises mobile device management (MDM). For more information, see Deploy Windows Installer apps to MDM-enrolled Windows devices .
Task sequence	Install or uninstall complex applications using task sequences. For more information, see Task sequence deployment type .

 **Note**

The Configuration Manager console may display other deployment types, but they are for platforms that are no longer supported. For more information, see [What happened to hybrid?.](#)

Note 1: Windows app package (in the Windows Store)

To deploy the app as a link to the Windows Store, configure the group policy **Turn off the Store application**. Set this policy to **Disabled** or **Not configured**. If you enable this setting, clients can't connect to the Windows Store to download and install applications.

Windows clients always evaluate deployment types that use a link to a store before other deployment types. Then the client evaluates deployment types by priority.

 **Tip**

Some store links may cause the following error in the Create Application Wizard: "Invalid Application link". For example, some store *Featured Apps* may cause this error. You can still select **Next** on the **General** page of the wizard. Configuration Manager successfully creates the app, and you can successfully deploy it.

Next steps

After creating an application in Configuration Manager, the next step is to [deploy the application](#).

Create a group of applications that you can send to a user or device collection as a single deployment. For more information, see [Create application groups](#).

For more information about creating applications on different OS platforms, see the following articles:

- [Create Windows applications](#)
- [Create Mac applications](#)
- [Create Windows Embedded applications](#)

Create Mac computer applications with Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Important

Starting in January 2022, this feature of Configuration Manager is deprecated. For more information, see [Mac computers](#).

Keep the following considerations in mind when you create and deploy applications for Mac computers.

Important

The procedures in this topic cover information about deploying applications to Mac computers on which you installed the Configuration Manager client. Mac computers that you enrolled with Microsoft Intune do not support application deployment.

General considerations

You can use Configuration Manager to deploy applications to Mac computers that run the Configuration Manager Mac client. The steps to deploy software to Mac computers are similar to the steps to deploy software to Windows computers. However, before you create and deploy applications for Mac computers that are managed by Configuration Manager, consider the following:

- Before you can deploy Mac application packages to Mac computers, you must use the **CMApUtil** tool on a Mac computer to convert these applications into a format that can be read by Configuration Manager.
- Configuration Manager does not support the deployment of Mac applications to users. Instead, these deployments must be made to a device. Similarly, for Mac application deployments, Configuration Manager does not support the **Pre-deploy software to the user's primary device** option on the **Deployment Settings** page of the **Deploy Software Wizard**.

- Mac applications support simulated deployments.
- You cannot deploy applications to Mac computers that have a purpose of **Available**.
- The option to send wake-up packets when you deploy software is not supported for Mac computers.
- Mac computers do not support Background Intelligent Transfer Service (BITS) for downloading application content. If an application download fails, it is restarted from the beginning.
- Configuration Manager does not support global conditions when you create deployment types for Mac computers.

Steps to create and deploy an application

The following table provides the steps, details, and information for creating and deploying applications for Mac computers.

Step	Details
Step 1: Prepare Mac applications for Configuration Manager	Before you can create Configuration Manager applications from Mac software packages, you must use the CMApUtil tool on a Mac computer to convert the Mac software into a Configuration Manager.cmmac file.
Step 2: Create a Configuration Manager application that contains the Mac software	Use the Create Application Wizard to create an application for the Mac software.
Step 3: Create a deployment type for the Mac application	This step is required only if you did not automatically import this information from the application.
Step 4: Deploy the Mac application	Use the Deploy Software Wizard to deploy the application to Mac computers.
Step 5: Monitor the deployment of the Mac application	Monitor the success of application deployments to Mac computers.

Supplemental procedures to create and deploy applications for Mac computers

Use the following procedures to create and deploy applications for Mac computers that are managed by Configuration Manager.

Step 1: Prepare Mac applications for Configuration Manager

The process for creating and deploying Configuration Manager applications to Mac computers is similar to the deployment process for Windows computers. However, before you create Configuration Manager applications that contain Mac deployment types, you must prepare the applications by using the **CMApUtil** tool. This tool is downloaded with the Mac client installation files. The **CMApUtil** tool can gather information about the application, which includes detection data from the following Mac packages:

- Apple disk image (.dmg)
- Meta package file (.mpkg)
- macOS X installer package (.pkg)
- macOS X application (.app)

After it gathers application information, the **CMApUtil** then creates a file with the extension **.cmmac**. This file contains the installation files for the Mac software and information about detection methods that can be used to evaluate whether the application is already installed. **CMApUtil** can also process **.dmg** files that contain multiple Mac applications and create different deployment types for each application.

1. Copy the Mac software installation package to the folder on the Mac computer where you extracted the contents of the **macclient.dmg** file that you downloaded from the Microsoft Download Center.
2. On the same Mac computer, open a terminal window and navigate to the folder where you extracted the contents of the **macclient.dmg** file.
3. Navigate to the **Tools** folder and type the following command-line command:

```
./CMApUtil <properties>
```

For example, say you want to convert the contents of an Apple disk image file named **MySoftware.dmg** that's stored in the user's desktop folder into a **cmmac** file in the same folder. You also want to create **cmmac** files for all applications that are found in the disk image file. To do this, use the following command line:

```
./CMApputil -c /Users/ <User Name> /Desktop/MySoftware.dmg -o /Users/ <User Name> /Desktop -a
```

ⓘ Note

The application name can't be more than 128 characters.

To configure options for **CMApputil**, use the command-line properties in the following table:

Property	More information
-h	Displays the available command-line properties.
-r	Outputs the detection.xml of the provided .cmmac file to stdout . The output contains the detection parameters and the version of CMApputil that was used to create the .cmmac file.
-c	Specifies the source file to be converted.
-o	Specifies the output path in conjunction with the -c property.
-a	Automatically creates .cmmac files in conjunction with the -c property for all applications and packages in the disk image file.
-s	Skips generating the detection.xml if no detection parameters are found and forces the creation of the .cmmac file without the detection.xml file.
-v	Displays more detailed output from the CMApputil tool together with diagnostic information.

4. Ensure that the **.cmmac** file has been created in the output folder that you specified.

Create a Configuration Manager application that contains the Mac software

Use the following procedure to help you create an application for Mac computers that are managed by Configuration Manager.

1. In the Configuration Manager console, choose **Software Library > Application Management > Applications**.
2. On the **Home** tab, in the **Create** group, choose **Create Application**.

3. On the **General** page of the **Create Application Wizard**, select **Automatically detect information about this application from installation files**.

 **Note**

If you want to specify information about the application yourself, select **Manually specify the application information**. For more information about how to manually specify the information, see [How to create applications with Configuration Manager](#).

4. In the **Type** drop-down list, select **Mac OS X**.

5. In the **Location** field, specify the UNC path in the form `\\<server>\<i><share>>\<i><filename>>` to the Mac application installation file (.cmmac file) that will detect application information. Alternatively, choose **Browse** to browse to and specify the installation file location.

 **Note**

You must have access to the UNC path that contains the application.

6. Choose **Next**.

7. On the **Import Information** page of the **Create Application Wizard**, review the information that was imported. If necessary, you can choose **Previous** to go back and correct any errors. Choose **Next** to proceed.

8. On the **General Information** page of the **Create Application Wizard**, specify information about the application such as the application name, comments, version, and an optional reference to help you reference the application in the Configuration Manager console.

 **Note**

Some of the application information might already be on this page if it was previously obtained from the application installation files.

9. Choose **Next**, review the application information on the **Summary** page, and then complete the **Create Application Wizard**.

10. The new application is displayed in the **Applications** node of the Configuration Manager console.

Step 3: Create a deployment type for the Mac application

Use the following procedure to help you create a deployment type for Mac computers that are managed by Configuration Manager.

ⓘ Note

If you automatically imported information about the application in the **Create Application Wizard**, a deployment type for the application might already have been created.

1. In the Configuration Manager console, choose **Software Library > Application Management > Applications**.
2. Select an application. Then, on the **Home** tab, in the **Application** group, choose **Create Deployment Type** to create a new deployment type for this application.

ⓘ Note

You can also start the **Create Deployment Type Wizard** from the **Create Application Wizard** and from the **Deployment Types** tab of the *<application name>* **Properties** dialog box.

3. On the **General** page of the **Create Deployment Type Wizard**, in the **Type** drop-down list, select **Mac OS X**.
4. In the **Location** field, specify the UNC path in the form `\\\<share>\<filename>` to the application installation file (.cmmac file). Alternatively, choose **Browse** to browse to and specify the installation file location.

ⓘ Note

You must have access to the UNC path that contains the application.

5. Choose **Next**.
6. On the **Import Information** page of the **Create Deployment Type Wizard**, review the information that was imported. If necessary, choose **Previous** to go back and

correct any errors. Choose **Next** to continue.

7. On the **General Information** page of the **Create Deployment Type Wizard**, specify information about the application such as the application name, comments, and the languages in which the deployment type is available.

 **Note**

Some of the deployment type information might already be on this page if it was previously obtained from the application installation files.

8. Choose **Next**.
9. On the **Requirements** page of the **Create Deployment Type Wizard**, you can specify the conditions that must be met before the deployment type can be installed on Mac computers.
10. Choose **Add** to open the **Create Requirement** dialog box and add a new requirement.

 **Note**

You can also add new requirements on the **Requirements** tab of the *<deployment type name> Properties* dialog box.

11. From the **Category** drop-down list, select that this requirement is for a device.
12. From the **Condition** drop-down list, select the condition that you want to use to assess whether the Mac computer meets the installation requirements. The contents of this list varies depending on the category that you select.
13. From the **Operator** drop-down list, choose the operator to use to compare the selected condition to the specified value to assess whether the user or device meets the installation requirements. The available operators vary depending on the selected condition.
14. In the **Value** field, specify the values to use with the selected condition and operator to assess whether the user or device meets in the installation requirement. The available values vary depending on the condition and operator that you select.
15. Choose **OK** to save the requirement rule and exit the **Create Requirement** dialog box.

16. On the Requirements page of the **Create Deployment Type Wizard**, choose **Next**.
17. On the **Summary** page of the **Create Deployment Type Wizard**, review the actions for the wizard to take. If necessary, choose **Previous** to go back and change deployment type settings. Choose **Next** to create the deployment type.
18. After the **Progress** page finishes, review the actions that have been taken, and then choose **Close** to complete the **Create Deployment Type Wizard**.
19. If you started this wizard from the **Create Application Wizard**, you will return to the **Deployment Types** page.

Deploy the Mac application

The steps to deploy an application to Mac computers are the same as the steps to deploy an application to Windows computers, except for the following differences:

- The deployment of applications to users is not supported.
- Deployments that have a purpose of **Available** are not supported.
- The **Pre-deploy software to the user's primary device** option on the **Deployment Settings** page of the **Deploy Software Wizard** is not supported.
- Because Mac computers do not support Software Center, the setting **User notifications** on the **User Experience** page of the **Deploy Software Wizard** is ignored.
- The option to send wake-up packets when you deploy software is not supported for Mac computers.

Note

You can build a collection that contains only Mac computers. To do so, create a collection that uses a query rule and use the example WQL query in the [How to create queries](#) topic.

For more information, see [Deploy applications](#).

Step 5: Monitor the deployment of the Mac application

You can use the same process to monitor application deployments to Mac computers as you would to monitor application deployments to Windows computers.

For more information, see [Monitor applications](#).

Create Windows applications in Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

In addition to the other Configuration Manager requirements and procedures for [creating an application](#), also take the following considerations into account when you create and deploy applications for Windows devices.

General considerations

Configuration Manager supports the deployment of Windows app package (`.appx`) and app bundle (`.appxbundle`) formats.

When you create an application in the Configuration Manager console, select the application installation file **Type** as **Windows app package (*.appx, *.appxbundle, *.msix, *.msixbundle)**. For more information on creating apps in general, see [Create applications](#). For more information on the MSIX format, see [Support for MSIX format](#).

Note

To take advantage of new Configuration Manager features, first update clients to the latest version. While new functionality appears in the Configuration Manager console when you update the site and console, the complete scenario isn't functional until the client version is also the latest.

Provision Windows app packages for all users on a device

Provision an application with a Windows app package for all users on the device. One common example of this scenario is provisioning an app from the Microsoft Store for Business and Education, like Minecraft: Education Edition, to all devices used by students in a school. Previously, Configuration Manager only supported installing these applications per user. After signing in to a new device, a student would have to wait to access an app. Now when the app is provisioned to the device for all users, they can be productive more quickly.

Important

Be careful with installing, provisioning, and updating different versions of the same Windows app package on a device, which may cause unexpected results. This behavior may occur when using Configuration Manager to provision the app, but then allowing users to update the app from the Microsoft Store. For more information, see the next step guidance when you [Manage apps from the Microsoft Store for Business](#).

When deploying offline apps to Windows devices with the Configuration Manager client, don't allow users to update applications external to Configuration Manager deployments. Control of updates to offline apps is especially important in multi-user environments such as classrooms. For more information, see [Manage apps from the Microsoft Store for Business and Education with Configuration Manager](#).

Configuration Manager supports app provisioning on all supported versions of Windows 10 and later.

To configure a Windows app deployment type for this feature, enable the option to **Provision this application for all users on the device**. For more information, see [Create applications](#).

Note

If you need to uninstall a provisioned application from devices to which users have already signed on, you need to create two uninstall deployments. Target the first uninstall deployment to a device collection that contains the devices. Target the second uninstall deployment to a user collection that contains the users who have already signed on to devices with the provisioned application. When uninstalling a provisioned app on a device, Windows currently doesn't uninstall that app for users as well.

Support for MSIX format

Configuration Manager supports the Windows app package (`.msix`) and app bundle (`.msixbundle`) formats. Supported versions of Windows 10 and later support these formats.

- For an overview of MSIX, see [A closer look at MSIX](#).

- For how to create a new MSIX app, see [MSIX support introduced in Insider Build 17682](#).

Convert applications to MSIX

Convert your existing Windows Installer (.msi) applications to the MSIX format.

Prerequisites for MSIX

- A reference device running Windows 10 version 1809 or later
- Sign in to Windows on this device as a user with local administrative rights
- Install the following apps on this device:
 - Configuration Manager console
 - Install the [MSIX Packaging Tool](#) from the Microsoft Store
 - Install the [MSIX packaging tool driver](#)

Don't install any other apps or services on this device. It's your reference system.

Process to convert applications to MSIX format

1. Elevate the Configuration Manager console, go to the **Software Library** workspace, expand **Application Management**, and select the **Applications** node.
2. Select an application that has a Windows Installer (.msi) deployment type.

ⓘ Note

You need to be able to access the application's source content from the reference device.

The application's name can't have any special characters. Configuration Manager uses the app name as the name of the output file.

Don't install this application on the reference device in advance.

3. Select **Convert to .MSIX** in the ribbon.

When the wizard completes, the MSIX Packaging Tool creates an MSIX file in the location you specified in the wizard. During this process, Configuration Manager silently

installs the application on the reference device.

If the process fails, the summary page points to the log file with more information. If there's an error about capturing user state, sign out of Windows. Signing in again may resolve this issue.

To use this MSIX app, you first need to digitally sign it so that clients trust it. For more information on this process, see the following articles:

- [MSIX - The MSIX Packaging Tool - signing the MSIX package](#)
- [How to sign an app package using SignTool](#)

After signing the app, create a new deployment type on the application in Configuration Manager. For more information, see [Create deployment types for the application](#).

Task sequence deployment type

ⓘ Note

In this version of Configuration Manager, the task sequence deployment type is a pre-release feature. To enable it, see [Pre-release features](#).

You can install complex applications using task sequences via the application model. Add a task sequence deployment type to an app either to install or uninstall the app. This deployment type provides the following behaviors:

- Display the app task sequence with an icon in Software Center. An icon makes it easier for users to find and identify the app task sequence.
- Define additional metadata for the app task sequence, including localized information
- Starting in version 2010, deploy an app task sequence to a user collection

You can only add a non-OS deployment task sequence as a deployment type on an app. High-impact, OS deployment, or OS upgrade task sequences aren't supported. A user-targeted deployment still runs in the context of the local System account.

When you add this deployment type to an app, configure its properties on the **Task Sequence** page. For more information, see [Deployment type Task Sequence options](#).

Starting in version 2006, use the following Windows PowerShell cmdlets to add and configure a task sequence deployment type:

- [Add-CMTaskSequenceDeploymentType](#)
- [Set-CMTaskSequenceDeploymentType](#)

ⓘ Note

Consider the following scenario:

- An application has a task sequence deployment type.
- It's deployed as available.
- A device has maintenance windows defined.
- A user on the device runs the deployment in Software Center outside of a maintenance window.

Configuration Manager honors the user's intent to install the application, even though there's no available maintenance window. In version 2107 and earlier, when the task sequence ran, the **Restart Computer** step would fail because of the maintenance window.

Starting in version 2111, this step now ignores maintenance windows only when the task sequence is run as an app deployment type.

Prerequisites for a task sequence deployment type

Create a custom task sequence:

- Use only non-OS deployment steps, for example: **Install Package**, **Run Command Line**, or **Run PowerShell Script**. For more information including the full list of supported steps, see [Create a task sequence for non-OS deployments](#).
- On the task sequence properties, **User Notification** tab, don't select the option for a high-impact task sequence.

When you create the application, to add a task sequence deployment type, your user account needs permission to read task sequences. Use one of the following options to configure these permissions:

- Add the app administrator's user account to the built-in **Read-Only Analyst** role. This role allows them to view all Configuration Manager objects.
- Copy the built-in **Application Administrator** role to create a custom role. Add the **Read** permission on the **Task Sequence Package** object.

Known issues for a task sequence deployment type

- Don't use the **Install Application** step in this task sequence. Use the [Install Package](#) step to install apps.
- In version 2006 and earlier, you can't yet deploy an app task sequence to a user collection. This issue was resolved in version 2010.

Support for Universal Windows Platform (UWP) apps

Windows 10 or later devices don't require a sideloading key to install line-of-business apps. To enable sideloading on Windows, however, the registry key

`HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\Appx\AllowAllTrustedApps`
must have a value of 1.

If you don't configure this registry key, Configuration Manager automatically sets this value to 1 the first time you deploy an app to the device. If you've set this value to 0, Configuration Manager can't automatically change the value, and your line-of-business app deployment fails.

Digitally sign UWP line-of-business apps. Use a code-signing certificate that's trusted on each device to which you deploy the app. Use certificates from your organization's PKI, or purchase a certificate from a third-party provider whose public root certificate is already trusted by Windows.

To sign mobile app packages, use the following table to determine the type of code-signing certificate to use:

Package	Symantec	Non-Symantec
Universal .appx packages on Windows 10 Mobile devices	Yes	Yes
.xap packages	Yes	No
.appx packages built for Windows Phone 8.1 to install on Windows 10 Mobile devices	Yes	No

Deploy Windows Installer apps to MDM-enrolled Windows 10 devices

The **Windows Installer through MDM (*.msi)** deployment type lets you create and deploy Windows Installer-based apps to MDM-enrolled devices running Windows 10 or later.

When you use this deployment type, consider the following points:

- Only upload a single file with the MSI extension.
- Configuration Manager uses the file's product code and product version for app detection.
- Windows uses the app's default restart behavior. Configuration Manager doesn't control the app restart behavior.
- Per-user MSI packages are installed for a single user.
- Per-machine MSI packages are installed for all users of the device.
- Configuration Manager supports app updates. The MSI product code of each version must be the same.

Create Windows Embedded applications with Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

In addition to the other Configuration Manager requirements and procedures for creating an application, you must also take the following considerations into account when you create and deploy applications for Windows Embedded devices.

General considerations

- When you deploy applications to Windows Embedded devices that are enabled for write filtering, you can specify whether to disable the write filter on the device during the app deployment. You can then choose to restart the write filter after the app deployment. If the write filter is not disabled, the software is deployed to a temporary overlay. This means that unless another deployment forces changes to persist, the software will no longer be installed when the device restarts.
- When you deploy an application to a Windows Embedded device, make sure that the device is a member of a collection that has a configured maintenance window. This lets you manage when the write filter is disabled and enabled, and when the device restarts.
- The setting that controls the write filter behavior is a check box named **Commit changes at deadline or during a maintenance window (requires restarts)**.

Tips for deploying applications

Use required applications rather than available applications for Windows Embedded devices that have write filters enabled. Because users cannot install apps from Software Center on a Windows Embedded device that has write filters enabled, always deploy applications with a deployment purpose of **required** rather than **available** to these devices. Typically, this isn't a problem because computers that run a Windows Embedded operating system often run a single application that must run in the same way for multiple users. Because of this, these devices are highly managed and locked down by the IT department. Required applications are well-suited to this scenario.

However, if users do run more than one application on embedded devices when write filters are enabled, educate these users about the following limitations:

- Users cannot install required software from Software Center.
- Users cannot change their business hours in the Options tab of Software Center.
- Users cannot postpone the installation of a required application.

In addition, low-rights users cannot log on during a maintenance period if Configuration Manager is committing changes for software installations and updates. During this period, users see a message informing them that the device is unavailable because it is being serviced.

Do not deploy applications to Windows Embedded devices that have write filters enabled if the applications require the user to accept the license terms. When write filters are disabled so that Configuration Manager can install software on embedded devices, low-rights users cannot log on to the device. If the installation requires the user to accept the license terms, this will not be possible and the installation will fail. Make sure that you do not deploy software to Windows Embedded devices if the installation requires user interaction. You can use the Applicable Platforms list to filter these operating systems.

How to create global conditions in Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

In Configuration Manager, global conditions are rules that represent business or technical conditions that you can use to specify how an application is provided and deployed to client devices. Global conditions are accessed from the Requirements page of the Create Deployment Type Wizard.

ⓘ Note

You can edit global conditions only from the site where they were created.

Use the following procedures to create Configuration Manager global conditions.

Provide basic information about the global condition

Several different types of global conditions are available. Different options are associated with the different global condition types. When you select a specific global condition type, Configuration Manager shows the options that apply to your selection.

1. In the Configuration Manager console, choose **Software Library > Application Management > Global Conditions**.
2. On the **Home** tab, in the **Create** group, choose **Create Global Condition**.
3. In the **Create Global Condition** dialog box, provide a name and an optional description for the global condition.
4. In the **Device type** drop-down list, choose whether the global condition is for a **Windows** computer or a **Windows Mobile** device.
5. In the **Condition Type** drop-down list, choose one of the following options:
 - **Setting** – This option checks for the existence of one or more items on client devices. For example, you can check that a file, folder, or registry key value exists on a client device.

- **Expression** – This option lets you to set up more complex rules to check if the condition is satisfied on client devices. For example, you can check if the physical memory on a computer is between 2 GB and 4 GB or if a mobile device uses touch screen input.

Set up rules for the global condition

The procedure to define the global condition rules is different depending on whether you are configuring a setting or an expression. Use the applicable procedure here to set up a setting or an expression for the global condition.

To set up a setting for the global condition

1. In the **Condition Type** drop-down list, choose **Setting**.
2. In the **Setting type** drop-down list, choose the item to use as the condition for which requirements will be checked. The following setting types and configurations are available.
 - **Active Directory query**
 - **LDAP prefix** - Specify a valid LDAP prefix to the Active Directory Domain Services query to assess compliance on client computers. You can use either **LDAP://** or **GC://**.
 - **Distinguished name (DN)** - Specify the distinguished name of the Active Directory Domain Services object that will be assessed for compliance on client computers.
 - **Search filter** - Specify an optional LDAP filter to refine the results from the Active Directory Domain Services query to assess compliance on client computers.
 - **Search scope** - Specify the search scope in Active Directory Domain Services:
 - **Base** - Queries only the specified object.
 - **One Level** - This option is not used in this version of Configuration Manager.
 - **Subtree** - Queries the specified object and its complete subtree in the directory.

- **Property** - Specify the property of the Active Directory Domain Services object that will be used to assess compliance on client computers.
 - **Query** - Shows the LDAP query that is constructed from the entries in **LDAP prefix**, **Distinguished name (DN)**, **Search Filter** if specified, and **Property**. This query will be used to assess compliance on client computers.
- **Assembly**
 - **Assembly name** - Specifies the name of the assembly object to search for. The name cannot be the same as any other assembly object of the same type, and the name must be registered in the Global Assembly Cache. The assembly name can be a maximum of 256 characters.

 **Note**

An assembly is a piece of code that can be shared between applications. Assemblies can have the .dll or .exe file name extension. The Global Assembly Cache is a folder named `%systemroot%\assembly` on client computers in which all shared assemblies are stored.

- **File system**

- **Type** – From the drop-down list, choose whether you want to search for a **File** or a **Folder**.
- **Path** - Specify the path to the specified file or folder on client computers. You can specify system environment variables and the `%USERPROFILE%` environment variable in the path.

 **Note**

If you use the `%USERPROFILE%` environment variable in the **Path** or **File or folder name** fields, all user profiles on the client computer will be searched. This could result in the discovery of multiple instances of the file or folder.

- **File or folder name** - Specify the name of the file or folder object that will be searched for. You can specify system environment variables and the `%USERPROFILE%` environment variable in the file or folder name. You can also use the * and ? wildcards in the file name.

Note

If you specify a file or folder name and use wildcards, this might produce a high numbers of results. This could result in high resource use on the client computer and high network traffic when reporting results to Configuration Manager.

- **Include subfolders** – Enable this option if you also want to search any subfolders under the specified path.
- **This file or folder is associated with a 64-bit application** - Choose whether the 64-bit system file location (%windir%\system32) should be searched in addition to the 32-bit system file location (%windir%\syswow64) on Configuration Manager clients that run a 64-bit version of Windows.

Note

If the same file or folder exists in both the 64-bit and 32-bit system file locations on the same 64-bit computer, multiple files will be discovered by the global condition.

The **File system** setting type does not support specifying a UNC path to a network share in the **Path** field.

- **IIS metabase**
 - **Metabase path** - Specify a valid path to the IIS Metabase.
 - **Property ID** - Specify the numeric property of the IIS Metabase setting.
- **Registry key**
 - **Hive** – From the drop-down list, choose the registry hive that you want to search in.
 - **Key** - Specify the registry key name that you want to search for. The format used should be *key\subkey*.
 - **This registry key is associated with a 64-bit application** - Specifies whether the 64-bit registry keys should be searched in addition to the 32-bit registry keys on clients that run a 64-bit version of Windows.

Note

If the same registry key exists in both the 64-bit and 32-bit registry locations on the same 64-bit computer, both registry keys will be discovered by the global condition.

- **Registry value**

- **Hive** - From the drop-down list, select the registry hive that you want to search in.
- **Key** - Specify the registry key name that you want to search for. The format used should be *key\subkey*.
- **Value** – Specify the value that must be contained within the specified registry key.
- **This registry key is associated with a 64-bit application** - Specifies whether the 64-bit registry keys should be searched in addition to the 32-bit registry keys on clients that run a 64-bit version of Windows.

Note

If the same registry key exists in both the 64-bit and 32-bit registry locations on the same 64-bit computer, both registry keys will be discovered by the global condition.

- **Script**

- **Discovery script** – Choose **Add** to enter, or browse to the script to use. You can use Windows PowerShell, VBScript, or JScript scripts.
- **Run scripts by using the logged on user credentials** – If you enable this option, the script will run on client computers by using the credentials of the user who is signed in.

Note

The value returned by the script will be used to assess the compliance of the global condition. For example, when you use VBScript, you could use the **WScript.Echo Result** command to return the Result variable value to the global condition.

If your script returns multiple values, these values must be on a single line and separated with a semi-colon. If each value is on a separate line, the evaluation will fail.

- **SQL query**
 - **SQL Server instance** – Choose whether you want the SQL query to run on the default instance, all instances, or a specified database instance name.

 **Note**

The instance name must refer to a local instance of SQL Server. To refer to a SQL Server Always On failover cluster instance or availability group, you should use a script setting.

- **Database** - Specify the name of the Microsoft SQL Server database for which the SQL query will be run.
- **Column** - Specify the column name returned by the Transact-SQL statement to use to assess the compliance of the global condition.
- **Transact-SQL statement** – Specify the full SQL query to use for the global condition. You can also choose **Open** to open an existing SQL query.

- **WQL query**

- **Namespace** - Specify the WMI namespace that will be used to build a WQL query that will be assessed for compliance on client computers. The default value is Root\cimv2.
- **Class** - Specifies the WMI class that will be used to build a WQL query that will be assessed for compliance on client computers.
- **Property** - Specifies the WMI property that will be used to build a WQL query that will be assessed for compliance on client computers.
- **WQL query WHERE clause** - You can use the **WQL query WHERE clause** item to specify a WHERE clause to be applied to the specified namespace, class, and property on client computers.

- **XPath query**

- **Path** - Specify the path to the XML file on client computers that will be used to assess compliance. Configuration Manager supports the use of all

Windows system environment variables and the %USERPROFILE% user variable in the path name.

- **XML file name** - Specify the file name that contains the XML query to use to assess compliance on client computers.
- **Include subfolders** - Enable this option if you also want to search any subfolders under the specified path.
- **This file is associated with a 64-bit application** - Choose whether the 64-bit system file location (%windir%\system32) should be searched in addition to the 32-bit system file location (%windir%\syswow64) on Configuration Manager clients that run a 64-bit version of Windows.
- **XPath query** - Specify a valid full XML path language (XPath) query to use to assess compliance on client computers.
- **Namespaces** - Opens the **XML Namespaces** dialog box to identify namespaces and prefixes to use during the XPath query.

3. In the **Data type** drop-down list, choose the format in which data will be returned by the condition before it is used to check requirements.

 **Note**

The **Data type** drop-down list is not shown for all setting types.

4. Set up further details about this setting below the **Setting type** drop-down list. The items you can set up will vary depending on the setting type you have selected.
5. Choose **OK** to save the rule and to close the **Create Global Condition** dialog box.

Set up an expression for the global condition

1. In the **Condition Type** drop-down list, choose **Expression**.
2. Choose **Add Clause** to open the **Add Clause** dialog box.
3. From the **Select category** drop-down list, select whether this expression is for a device or a user. Alternatively, select **Custom** to use a previously configured global condition.
4. From the **Select a condition** drop-down list, select the condition to use to assess whether the user or device meets the rule requirements. The contents of this list

will vary depending on the selected category.

5. From the **Choose operator** drop-down list, choose the operator that will be used to compare the selected condition to the specified value to assess whether the user or device meets the rule requirements. The available operators will vary depending on the selected condition.
6. In the **Value** field, specify the values that will be used with the selected condition and operator to assess whether the user or device meets the rule requirements. The available values will vary depending on the selected condition and the selected operator.
7. Choose **OK** to save the expression and to close the **Add Clause** dialog box.
8. When you have finished adding clauses to the global condition, choose **OK** to close the **Create Global Condition** dialog box and to save the global condition.

Create application groups

Article • 02/22/2023

Applies to: Configuration Manager (current branch)

Create a group of applications that you can send to a user or device collection as a single deployment. The metadata you specify about the app group is seen in Software Center as a single entity. You can order the apps in the group so that the client installs them in a specific order.

💡 Tip

This feature was first introduced in version 1906 as a **pre-release feature**. Beginning with version 2111, it's no longer a pre-release feature.

This feature is optional in Configuration Manager, and enabled by default. For more information, see **Enable optional features from updates**.

Process

1. In the Configuration Manager console, go to the **Software Library** workspace. Expand **Application Management** and select the **Application Group** node.
2. In the Create group in the ribbon, select **Create Application Group**.
3. On the **General Information** page, specify information about the app group.
4. On the **Software Center** page, include information that shows in Software Center.
5. On the **Application Group** page, select **Add**. Select one or more apps for this group. Reorder them using the **Move Up** and **Move Down** actions.
6. Complete the wizard.

💡 Tip

To manage app groups, you need permissions on the **Application Groups** object. The permissions for most administrative operations are the same as on applications.

Deploy

Deploy the app group using the same process as for an application. For more information, see [Deploy applications](#). You can deploy an app group to device or user collections. Starting in version 2111, when you deploy an app group as required to a device or user collection, you can specify that it automatically uninstalls when the resource is removed from the collection. For more information, see [Implicit uninstall](#).

After you deploy the group:

- If you add a new app to the group, you have to separately distribute the new app content to distribution points.
- If you modify an app in the app group, redistribute the content.

To troubleshoot an app group deployment, use the following log files on the client:

- **AppGroupHandler.log**
- **AppEnforce.log**
- **SettingsAgent.log**

App approval

Starting in version 2111, you can use the following [app approval](#) behaviors:

- Deploy an app group to a user collection and require approval.
 - A user can then request the app group in Software Center.
 - You can approve or deny the user's request for the app group.
- Deploy an app group to a device collection and require approval. The deployment is suspended on the device until you trigger installation via automation. For example, use the [Approve-CMAApprovalRequest](#) PowerShell cmdlet.
- From the Configuration Manager console, when you select a device, there's a new action in the **Device** group of the ribbon to **Install Application Group**. For more information, see [Install applications for a device](#).
- When you enable tenant attach, you can view status and take actions on app groups from the Microsoft Intune admin center. For more information, see [Install an application from the admin center](#).

Known issues

- The following deployment options may not work: alerts, phased deployment, repair.
- You can't use application groups with the **Install Application** task sequence step.
- You can't export or import app groups.
- In version 2103 and earlier, don't include in the group any apps that require restart, or the group deployment may fail.
- In version 2107 and earlier, if you delete an app that's a part of an app group, you'll see the following warning when you next view the properties of the app group: "Unable to load information about all applications in the group." Make a small change to the app group and save it. For example, add a space to the **Administrator comments**. When you save the change, it removes the deleted app from the group. Starting in version 2111, you can't delete an app that's part of an app group.
- In most scenarios, user categories on the app group don't display as filters in Software Center. If the app group is deployed as available to a user collection, the categories display.

PowerShell

You can create and deploy app groups using Windows PowerShell. For more information, see the following cmdlet articles:

- [Get-CMApplicationGroup](#)
- [New-CMApplicationGroup](#)
- [Remove-CMApplicationGroup](#)
- [Set-CMApplicationGroup](#)
- [Get-CMApplicationGroupDeployment](#)
- [New-CMApplicationGroupDeployment](#)
- [Remove-CMApplicationGroupDeployment](#)
- [Set-CMApplicationGroupDeployment](#)

Next steps

[Deploy applications](#)

Packages and programs in Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Configuration Manager continues to support packages and programs that were used in Configuration Manager 2007. A deployment that uses packages and programs might be more suitable than an application when you deploy any of the following tools or scripts:

- Administrative tools that don't install an application on a computer
- "One-off" scripts that don't need to be continually monitored
- Scripts that run on a recurring schedule and can't use global evaluation

💡 Tip

Consider using the **Scripts** feature in the Configuration Manager console. Scripts may be a better solution for some of the preceding scenarios instead of using packages and programs.

When you migrate packages from an earlier version of Configuration Manager, you can deploy them in your Configuration Manager hierarchy. After migration is complete, the packages appear in the **Packages** node in the **Software Library** workspace.

You can modify and deploy these packages in the same way you did by using software distribution. The **Import Package from Definition Wizard** remains in Configuration Manager to import legacy packages. Advertisements are converted to deployments when you migrate from Configuration Manager 2007 to a Configuration Manager hierarchy.

ⓘ Note

Use Package Conversion Manager to convert packages and programs into Configuration Manager applications. Package Conversion Manager is integrated with Configuration Manager. For more information, see **Package Conversion Manager**.

Packages can use some new features of Configuration Manager, including distribution point groups and monitoring. You can't deploy Microsoft Application Virtualization (App-V) applications with packages and programs in Configuration Manager. To

distribute virtual applications, create them as Configuration Manager applications. For more information, see [Deploy App-V virtual applications](#).

Create a package and program

Use the Create Package and Program wizard

1. In the Configuration Manager console, go to the **Software Library** workspace, expand **Application Management**, and select the **Packages** node.
2. In the **Home** tab of the ribbon, in the **Create** group, choose **Create Package**.
3. On the **Package** page of the **Create Package and Program Wizard**, specify the following information:
 - **Name:** Specify a name for the package with a maximum of 50 characters.
 - **Description:** Specify a description for this package with a maximum of 128 characters.
 - **Manufacturer** (optional): Specify a manufacturer name to help you identify the package in the Configuration Manager console. This name can be a maximum of 32 characters.
 - **Language** (optional): Specify the language version of the package with a maximum of 32 characters.
 - **Version** (optional): Specify a version number for the package with a maximum of 32 characters.
 - **This package contains source files:** This setting indicates whether the package requires source files to be present on client devices. By default, the wizard doesn't enable this option, and Configuration Manager doesn't use distribution points for the package. When you select this option, specify the package content to distribute to distribution points.
 - **Source folder:** If the package contains source files, choose **Browse** to open the **Set Source Folder** dialog box, and then specify the location of the source files for the package.

 **Note**

The computer account of the site server must have read access permissions to the source folder that you specify.

Windows limits the source path to 256 characters or less. This limit applies to package source as well as applications. For more information, see [Naming Files, Paths, and Namespaces](#).

- If you want to pre-cache content on a client, specify the **Architecture** and **Language** of the package. For more information, see [Configure pre-cache content](#).
4. On the **Program Type** page of the [Create Package and Program Wizard](#), select the **Standard** program type for computers. Or you can skip this step and create a program later.

Tip

To create a new program for an existing package, first select the package. Then, in the **Home** tab, in the **Package** group, choose **Create Program** to open the [Create Program Wizard](#).

The **Program for device** type is a legacy option that only applies to mobile devices, which aren't currently managed by Configuration Manager.

Custom icons for packages

Starting in version 2203, add custom icons for packages. These icons appear in Software Center when you deploy the package and program. Instead of a default icon, a custom icon can improve the user experience to better identify the software.

On the **General** tab of package properties, in the section for the icon, select **Browse**. Select an icon from the default shell library, or browse to another file in a local or network path.

- It supports the following file types:
 - Programs (.exe)
 - Libraries (.dll)
 - Icons (.ico)
 - Images (.png, .jpeg, .jpg)
- The file doesn't need to be on clients that you target with the deployment. Configuration Manager includes the image with the deployment policy.

- The maximum file size for an image is 256 KB.
- Icons can have pixel dimensions of up to 512 x 512.

When clients receive the deployment policy, they'll display the icon in Software Center.

 **Note**

To take full advantage of new Configuration Manager features, after you update the site, also update clients to the latest version. While new functionality appears in the Configuration Manager console when you update the site and console, the complete scenario isn't functional until the client version is also the latest.

Create a program

1. On the **Program Type** page of the **Create Package and Program Wizard**, choose **Standard Program**, and then choose **Next**.
2. On the **Standard Program** page, specify the following information:
 - **Name:** Specify a name for the program with a maximum of 50 characters.

 **Note**

The program name must be unique within a package. After you create a program, you can't modify its name.

- **Command Line:** Enter the command line to use to start this program, or choose **Browse** to browse to the file location.

If you don't specify an extension for a file name, Configuration Manager attempts to use .com, .exe, and .bat as possible extensions.

When the client runs the program, Configuration Manager searches for the file in the following locations:

- Within the package
- The local Windows folder
- The local %path%

If it can't find the file, the program fails.

- **Startup folder** (optional): Specify the folder from which the program runs, up to 127 characters. This folder can be an absolute path on the client. It can

also be a path that's relative to the distribution point folder that contains the package.

- **Run:** Specify the mode in which the program runs on client computers. Select one of the following options:
 - **Normal:** The program runs in the normal mode based on system and program defaults. This mode is the default.
 - **Minimized:** The program runs minimized on client devices. Users might see installation activity in the notification area or on the taskbar.
 - **Maximized:** The program runs maximized on client devices. Users see all installation activity.
 - **Hidden:** The program runs hidden on client devices. Users don't see any installation activity.
- **Program can run:** Specify whether the program runs only when a user is signed in, only when no user is signed in, or regardless of whether a user is signed in to the client computer.
- **Run mode:** Specify whether the program runs with administrative permissions or with the permissions of the user who's currently signed in.
- **Allow users to view and interact with the program installation:** Use this setting, if available, to specify whether to allow users to interact with the program installation. This option is only available if the following conditions are met:
 - **Program can run** setting is **Only when a user is logged on** or **Whether or not a user is logged on**
 - **Run mode** setting is to **Run with administrative rights**
- **Drive mode:** Specify information about how this program runs on the network. Choose one of the following options:
 - **Runs with UNC name:** Specify that the program runs with a Universal Naming Convention (UNC) name. This setting is the default.
 - **Requires drive letter:** Specify that the program requires a drive letter to fully qualify its location. For this setting, Configuration Manager can use any available drive letter on the client. This setting requires the deployment to use the Deployment option **Run program from distribution point** and the package's Data Access option enabled to **Copy the content in this package to a package share on distribution points**.

- **Requires specific drive letter:** Specify that the program requires a specific drive letter that you specify to fully qualify its location. For example, Z:. If the client is already using the specified drive letter, the program doesn't run. This setting requires the deployment to use the Deployment option **Run program from distribution point** and the package's Data Access option enabled to **Copy the content in this package to a package share on distribution points**.
- **Reconnect to distribution point at log on:** Indicate whether the client reconnects to the distribution point when the user signs in. By default, the wizard doesn't enable this option.

3. On the **Requirements** page of the **Create Package and Program Wizard**, specify the following information:

- **Run another program first:** Identify a package and program that runs before this package and program runs.
- **Platform requirements:** Select **This program can run on any platform** or **This program can run only on specified platforms**. Then choose the OS versions that clients must have to install this package and program.
- **Estimated disk space:** Specify the amount of disk space that the program requires to run on the computer. The default setting is **Unknown**. If necessary, specify a whole number greater than or equal to zero. If you set a value, also select units for the value.
- **Maximum allowed run time (minutes):** Specify the maximum time that you expect the program to run on the client computer. The default value is **120** minutes. Only use whole numbers greater than zero.

Important

If you use maintenance windows on the same collection to which you deploy this program, a conflict could occur if the **Maximum allowed run time** is longer than the scheduled maintenance window. If you set the maximum run time to **Unknown**, the program starts to run during the maintenance window. It then continues to run as needed after the maintenance window is closed. If you set the maximum run time to a specific period that's greater than the length of any available maintenance window, then the client doesn't run the program.

If you set this value to **Unknown**, Configuration Manager sets the maximum allowed run time as 12 hours (720 minutes).

ⓘ Note

If the program exceeds the maximum run time, Configuration Manager stops it if the following conditions are met:

- You enable the option to **Run with administrative rights**
- You don't enable the option to **Allow users to view and interact with the program installation**

Deploy packages and programs

1. In the Configuration Manager console, go to the **Software Library** workspace, expand **Application Management**, and select the **Packages** node.
2. Select the package that you want to deploy. In the **Home** tab of the ribbon, in the **Deployment** group, choose **Deploy**.
3. On the **General** page of the **Deploy Software Wizard**, specify the name of the package and program that you want to deploy. Select the collection to which you want to deploy the package and program, and any optional comments.

To store the package content on the collection's default distribution point group, select the option to **Use default distribution point groups associated to this collection**. If you didn't associate this collection with a distribution point group, this option is unavailable.

4. On the **Content** page, choose **Add**. Select the distribution points or distribution point groups to which you want to distribute the content for this package and program.
5. On the **Deployment Settings** page, configure the following settings:
 - **Purpose:** Choose one of the following options:
 - **Available:** The user sees the published package and program in Software Center and can install it on demand.
 - **Required:** The package and program is deployed automatically, according to the configured schedule. In Software Center, you can track its deployment status and install it before the deadline.

 **Note**

If multiple users are signed into the device, package and task sequence deployments may not appear in Software Center.

- **Send wake-up packets:** If you set the deployment purpose to **Required** and select this option, the site first sends a wake-up packet to computers at the installation deadline time. Before you can use this option, configure computers for Wake On LAN. For more information, see [How to configure Wake on LAN](#).
- **Allow clients on a metered Internet connection to download content after the installation deadline, which might incur additional costs**

 **Note**

When you deploy a package and program, the option to **Pre-deploy software to the user's primary device** isn't available.

6. On the **Scheduling** page, configure when to deploy this package and program to client devices.

The options on this page vary depending on whether you set the deployment action to **Available** or **Required**.

For **Required** deployments, configure the rerun behavior for the program from the **Rerun behavior** drop-down menu. Choose from the following options:

Rerun behavior	Description
Never rerun deployed program	The client won't rerun the program. This behavior happens even if the program originally failed or if the program files are changed.
Always rerun program	The client always reruns the program when the deployment is scheduled. This behavior happens even if the program has already successfully run. It's useful with recurring deployments when you update the program.
Rerun if failed previous attempt	The client reruns the program when the deployment is scheduled, only if it failed on the previous run attempt.

Rerun behavior	Description
Rerun if succeeded on previous attempt	The client reruns the program only if it previously ran successfully on the client. This behavior is useful with recurring deployments when you routinely update the program, and each update requires the previous update to be successfully installed.

7. On the **User Experience** page, specify the following information:

- **Allow users to run the program independently of assignments:** Users can install this software from Software Center regardless of any scheduled installation time.
- **Software installation:** Allows the software to be installed outside of any configured maintenance windows.
- **System restart (if required to complete the installation):** If the software installation requires a device restart to finish, allow this action to happen outside of any configured maintenance windows.
- **Embedded devices:** When you deploy packages and programs to Windows Embedded devices that are write-filter-enabled, you can specify that they install packages and programs on the temporary overlay and commit changes later. Alternately, commit the changes on the installation deadline or during a maintenance window. When you commit changes on the installation deadline or during a maintenance window, a restart is required, and the changes persist on the device.

(!) Note

When you deploy a package or program to a Windows Embedded device, make sure that the device is a member of a collection that has a configured maintenance window. For more information about how maintenance windows are used when you deploy packages and programs to Windows Embedded devices, see [Creating Windows Embedded applications](#).

8. On the **Distribution Points** page, specify the following information:

- **Deployment options:** Specify the action that a client takes when it uses a distribution point in its current boundary group. Also select the action for the

client when it uses a distribution point from a neighbor boundary group or the default site boundary group.

Important

If you configure the deployment option to **Run program from distribution point**, make sure to enable the option to **Copy the content in this package to a package share on distribution points** on the **Data Access** tab of the package properties. Otherwise the package is unavailable to run from distribution points.

- **Allow clients to use distribution points from the default site boundary group:** When this content isn't available from any distribution point in the current or neighbor boundary groups, enable this option to let them try distribution points in the site default boundary group.

9. Complete the wizard.

View the deployment in the **Deployments** node of the **Monitoring** workspace and in the details pane of the package deployment tab when you select the deployment. For more information, see [Monitor packages and programs](#).

Monitor packages and programs

To monitor package and program deployments, use the same procedures that you use to monitor applications as detailed in [Monitor applications](#).

Packages and programs also include a number of built-in reports, which enable you to monitor information about the deployment status of packages and programs. These reports have the report category of **Software Distribution - Packages and Programs** and **Software Distribution - Package and Program Deployment Status**.

For more information about how to configure reporting in Configuration Manager, see [Introduction to reporting](#).

Manage packages and programs

In the **Software Library** workspace, expand **Application Management**, and select the **Packages** node. Select the package that you want to manage, and then choose a management task.

Create Prestage Content File

Opens the **Create Prestaged Content File Wizard**, to create a file that contains the package content. Use this file to manually import the package to a remote distribution point. This action is useful when you have low network bandwidth between the site server and the distribution point.

Create Program

Opens the **Create Program Wizard**, to create a new program for this package.

Export

Opens the **Export Package Wizard**, to export the selected package and its content to a file. Use this file to import the file to another hierarchy.

Deploy

Opens the **Deploy Software Wizard**, to deploy the selected package and program to a collection. For more information, see [Deploy packages and programs](#).

Distribute content

Opens the **Distribute Content Wizard**, to send the content for a package and program to selected distribution points or distribution point groups.

Import

Opens the **Import Package Wizard**, to import a previously exported package from a .zip file.

💡 Tip

When you import an object in the Configuration Manager console, it imports to the current folder. In earlier versions, Configuration Manager always put imported objects in the root node.

Update distribution points

Updates distribution points with the latest content for the selected package and program.

Next steps

- [Scripts](#)
- [Package Conversion Manager](#)
- [Package definition files](#)

Package definition files

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Package definition files are scripts to help you automate the creation of [Packages and programs](#) in Configuration Manager. They provide all of the information that Configuration Manager needs to create a package and program, except for the location of package source files.

About the package definition file format

Each package definition file is an ASCII or UTF-8 text file that uses the .ini file format. It contains the following sections:

[PDF]

This section identifies the file as a package definition file. It contains the following information:

- **Version:** Specify the version of the package definition file format that the file uses. This version corresponds to the version of Configuration Manager for which it was written. This entry is required.

[Package Definition]

Specify the properties of the package and program. It provides the following information:

- **Name:** The name of the package, up to 50 characters.
- **Version (optional):** The version of the package, up to 32 characters.
- **Icon (optional):** The file that contains the icon to use for this package. If specified, this icon replaces the default package icon in the Configuration Manager console.
- **Publisher:** The publisher of the package, up to 32 characters.
- **Language:** The language version of the package, up to 32 characters.
- **Comment (optional):** A comment about the package, up to 127 characters.

- **ContainsNoFiles**: This entry indicates if the package has any source files.
- **Programs**: The programs that you define for this package. Each program name corresponds to a **[Program]** section in this package definition file.

Example:

```
Programs=Typical, Custom, Uninstall
```

- **MIFFileName**: The name of the Management Information Format (MIF) file that contains the package status, up to 50 characters.
- **MIFName**: The name of the package for MIF matching, up to 50 characters.
- **MIFVersion**: The version number of the package for MIF matching, up to 32 characters.
- **MIFPublisher**: The software publisher of the package for MIF matching, up to 32 characters.

[Program]

Include a **[Program]** section for each program that you specify in the **Programs** entry in the **[Package Definition]** section. This section defines each program. Each program section provides the following information:

- **Name**: The name of the program, up to 50 characters. This entry must be unique within a package.
- **Icon** (optional): Specify the file that contains the icon to use for this program. This icon replaces the default program icon in the Configuration Manager console. The client also displays this icon when you deploy the program to a collection.
- **Comment** (optional): A comment about the program, up to 127 characters.
- **CommandLine**: Specify the command line for the program, up to 127 characters. The command is relative to the package source folder.
- **StartIn**: Specify the working folder for the program, up to 127 characters. This entry can be an absolute path on the client computer or a path that's relative to the package source folder.
- **Run**: Specify the program mode in which the program runs. You can specify **Minimized**, **Maximized**, or **Hidden**. If you don't include this entry, the program runs in normal mode.

- **AfterRunning:** Specify any special action that occurs after the program successfully completes. Options available are **SMSRestart**, **ProgramRestart**, or **SMSLogoff**. If you don't include this entry, the program doesn't run a special action.
- **EstimatedDiskSpace:** Specify the amount of disk space that the software program requires to run on the computer. The default value is **Unknown**. You can set the value as a whole number greater than or equal to zero. If you specify a value, also include the units for the value.

Example:

```
EstimatedDiskSpace=38MB
```

- **EstimatedRunTime:** Specify the estimated duration in minutes that you expect the program to run on the client computer. The default value is **120**. You can set the value as a whole number greater than zero, or **Unknown**.

Example:

```
EstimatedRunTime=25
```

- **SupportedClients:** Specify the processors and operating systems on which this program runs. Separate the platforms by commas. If you don't include this entry, the client doesn't check supported platforms for this program.
- **SupportedClientMinVersionX**, **SupportedClientMaxVersionX**: Specify the beginning-to-ending range for version numbers for the operating systems that are specified in the **SupportedClients** entry.

Example:

INI

```
SupportedClients=Win NT (I386),Win NT (IA64),Win NT (x64)
Win NT (I386) MinVersion1=5.00.2195.4
Win NT (I386) MaxVersion1=5.00.2195.4
Win NT (I386) MinVersion2=5.10.2600.2
Win NT (I386) MaxVersion2=5.10.2600.2
Win NT (I386) MinVersion3=5.20.0000.0
Win NT (I386) MaxVersion3=5.20.9999.9999
Win NT (I386) MinVersion4=5.20.3790.0
Win NT (I386) MaxVersion4=5.20.3790.2
Win NT (I386) MinVersion5=6.00.0000.0
Win NT (I386) MaxVersion5=6.00.9999.9999
Win NT (IA64) MinVersion1=5.20.0000.0
Win NT (IA64) MaxVersion1=5.20.9999.9999
Win NT (x64) MinVersion1=5.20.0000.0
Win NT (x64) MaxVersion1=5.20.9999.9999
```

```
Win NT (x64) MinVersion2=5.20.3790.0
Win NT (x64) MaxVersion2=5.20.9999.9999
Win NT (x64) MinVersion3=5.20.3790.0
Win NT (x64) MaxVersion3=5.20.3790.2
Win NT (x64) MinVersion4=6.00.0000.0
Win NT (x64) MaxVersion4=6.00.9999.9999
```

- **AdditionalProgramRequirements** (optional): Provide any other information or requirements for client computers, up to 127 characters.
- **CanRunWhen**: Specify the user status that the program requires to run on the client computer. Available values are **UserLoggedOn**, **NoUserLoggedOn**, or **AnyUserStatus**. The default value is **UserLoggedOn**.
- **UserInputRequired**: Specify whether the program requires interaction with the user. Available values are **True** or **False**. The default value is **True**. This entry is set to **False** if **CanRunWhen** isn't set to **UserLoggedOn**.
- **AdminRightsRequired**: Specify whether the program requires administrative credentials on the computer to run. Available values are **True** or **False**. The default value is **False**. This entry is set to **True** if **CanRunWhen** isn't set to **UserLoggedOn**.
- **UseInstallAccount**: Specify whether the program uses the client software installation account when it runs on client computers. By default, this value is **False**. This value is also **False** if **CanRunWhen** is set to **UserLoggedOn**.
- **DriveLetterConnection**: Specify whether the program requires a drive letter connection to the package files on the distribution point. You can specify **True** or **False**. The default value is **False**, which enables the program to use a Universal Naming Convention (UNC) connection. When this value is set to **True**, the client uses the next available drive letter, starting with Z: and proceeding backwards.
- **SpecifyDrive** (optional): Specify a drive letter that the program requires to connect to the package files on the distribution point. This setting forces the use of the specified drive letter for client connections to distribution points.
- **ReconnectDriveAtLogon**: Specify whether the computer reconnects to the distribution point when the user signs in. Available values are **True** or **False**. The default value is **False**.
- **DependentProgram**: Specify a program in this package that must run before the current program. This entry uses the format `DependentProgram=<ProgramName>`, where `<ProgramName>` is the **Name** entry for that program in the package definition file. If there are no dependent programs, leave this entry empty.

Examples:

```
DependentProgram=Admin
```

```
DependentProgram=
```

- **Assignment:** Specify how the program is assigned to users. This value can be:
 - **FirstUser:** Only the first user who signs in to the client runs the program
 - **EveryUser:** Every user who signs in runs the program
- When **CanRunWhen** isn't set to **UserLoggedOn**, this entry is set to **FirstUser**.
- **Disabled:** Specify whether you can deploy this program to clients. Available values are **True** or **False**. The default value is **False**.

Use a package definition file

1. In the Configuration Manager console, go to the **Software Library** workspace, expand **Application Management**, and select the **Packages** node.
2. On the **Home** tab of the ribbon, in the **Create** group, choose **Create Package from Definition**.
3. On the **Package Definition** page of the **Create Package from Definition Wizard**, choose an existing package definition file. To open a new package definition file, choose **Browse**. After you specify a new package definition file, select it from the **Package definition** list.
4. On the **Source Files** page, specify information about any required source files for the package and program.
5. If the package requires source files, on the **Source Folder** page, specify the location from where the site can get the source files.
6. Complete the wizard.

See also

[Packages and programs](#)

Deploy applications with Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Create or simulate a deployment of an application to a device or user collection in Configuration Manager. This deployment gives instructions to the Configuration Manager client on how and when to install or uninstall the software.

Before you can deploy an application, create at least one *deployment type* for the application. For more information, see [Create deployment types for the application](#).

In some situations, consider another feature as a better solution:

- If you have several applications that you need to deploy together, instead of creating multiple deployments, create an *application group*. You can send the app group to a user or device collection as a single deployment. For more information, see [Create application groups](#).
- For more complex deployments, first test it with a *simulated deployment*. This simulation tests the applicability of a deployment without installing or uninstalling the application. A simulated deployment evaluates the detection method, requirements, and dependencies for a deployment type and reports the results in the **Deployments** node of the **Monitoring** workspace. For more information, see [Simulate application deployments](#).

Note

You can only simulate the deployment of required applications, but not packages or software updates.

On-prem MDM-enrolled devices don't support simulated deployments, user experience, or scheduling settings.

- *Phased deployments* allow you to orchestrate a coordinated, sequenced rollout of software based on customizable criteria and groups. For example, deploy the application to a pilot collection, and then automatically continue the rollout based on success criteria. For more information, see [Create a phased deployment](#).

Start the deployment wizard

1. In the Configuration Manager console, go to the **Software Library** workspace, expand **Application Management**, and select either the **Applications** or **Application Groups** node.
2. Select an application or application group from the list to deploy. In the ribbon, select **Deploy**.

Note

When you view the properties of an existing deployment, the following sections correspond to tabs of the deployment properties window:

- General
- Content
- Deployment Settings
- Scheduling
- User Experience
- Alerts

General information

On the **General** page of the Deploy Software wizard, specify the following information:

- **Software:** This value displays the application to deploy. Select **Browse** to choose a different application.
- **Collection:** Select **Browse** to choose the target collection for this application deployment.
- **Use default distribution point groups associated to this collection:** Store the application content on the collection's default distribution point group. If you haven't associated the selected collection with a distribution point group, this option is grayed out.
- **Automatically distribute content for dependencies:** If any of the deployment types in the application have dependencies, then the site also sends dependent application content to distribution points.

Note

If you update the dependent application after deploying the primary application, the site doesn't automatically distribute any new content for the dependency.

- **Comments (optional):** Optionally, enter a description for this deployment.

Content options

On the **Content** page, select **Add** to distribute the content for this application to a distribution point or a distribution point group.

If you selected the option to **Use default distribution points associated to this collection** on the General page, then this option is automatically populated. Only a member of the **Application Administrator** security role can modify it.

If the application content is already distributed, then they appear here.

Deployment settings

On the **Deployment Settings** page, specify the following information:

- **Action:** From the drop-down list, choose whether this deployment is to **Install** or **Uninstall** the application.

ⓘ Note

If you create a deployment to **Install** an app and another deployment to **Uninstall** the same app on the same device, the **Install** deployment takes priority.

You can't change the action of a deployment after you create it.

- **Purpose:** From the drop-down list, choose one of the following options:
 - **Available:** The user sees the application in Software Center. They can install it on demand.

ⓘ Note

When you deploy apps as available to user collections, there are other requirements for some types of clients. For more information, see

Prerequisites to deploy user-available apps.

- **Required:** The client automatically installs the app according to the schedule that you set. If the application isn't hidden, a user can track its deployment status. They can also use Software Center to install the application before the deadline.

Note

When you set the deployment action to **Uninstall**, the deployment purpose is automatically set to **Required**. You can't change this behavior.

- **Allow end users to attempt to repair this application:** If you created the application with a repair command line, enable this option. Users see an option in Software Center to **Repair** the application.
- **Uninstall this application if the targeted object falls out of the collection:** Starting in version 2107, when you remove the device from the target collection, Configuration Manager runs the uninstall program on that device. For more information, see [Implicit uninstall](#). This option is only available for device-targeted deployments and when the deployment is **Required**.
- **Pre-deploy software to the user's primary device:** If the deployment is to a user, select this option to deploy the application to the user's primary device. This setting doesn't require the user to sign in before the deployment runs. If the user must interact with the installation, don't select this option. This option is only available when the deployment is **Required**.
- **Send wake-up packets:** If the deployment is **Required**, Configuration Manager sends a wake-up packet to computers before the client runs the deployment. This packet wakes the computers at the installation deadline time. Before using this option, computers and networks must be configured for Wake On LAN. For more information, see [Plan how to wake up clients](#).
- **Allow clients on a metered Internet connection to download content after the installation deadline, which might incur additional costs:** This option is only available for deployments with a purpose of **Required**.
- **Automatically upgrade any superseded versions of this application:** The client upgrades any superseded version of the application with the superseding application.

Note

This option works regardless of administrator approval. If an administrator already approved the superseded version, they don't need to also approve the superseding version. Approval is only for new requests, not superseding upgrades.

For **Available** install purpose, you can enable or disable this option.

Approval settings

The application approval behavior depends upon whether you enable the recommended optional feature, **Approve application requests for users per device**.

- **An administrator must approve a request for this application on the device:** If you enable the optional feature, the administrator approves any user requests for the application before the user can install it on the requested device. If the administrator approves the request, the user is only able to install the application on that device. The user must submit another request to install the application on another device. This option is grayed out when the deployment purpose is **Required**, or when you deploy the application to a device collection.
- **Require administrator approval if users request this application:** If you don't enable the optional feature, the administrator approves any user requests for the application before the user can install it. This option is grayed out when the deployment purpose is **Required**, or when you deploy the application to a device collection.

For more information, see [Approve applications](#).

Deployment properties: Deployment settings

When you view the properties of a deployment, if supported by the deployment type technology, the following option appears on the **Deployment Settings** tab:

Automatically close any running executables you specified on the install behavior tab of the deployment type properties dialog box. For more information, see [check for running executable files before installing an application](#).

Scheduling settings

On the **Scheduling** page, set the time when this application is deployed or available to client devices.

By default, Configuration Manager makes the deployment policy available to clients right away. If you want to create the deployment, but not make it available to clients until a later date, configure the option to **Schedule the application to be available**. Then select the date and time, including whether that's based on UTC or the client's local time.

If the deployment is **Required**, also specify the **Installation deadline**. By default this deadline is as soon as possible.

For example, you need to deploy a new line-of-business application. All users need to install it by a certain time, but you want to give them the option to opt in early. You also need to make sure that the site has distributed the content to all distribution points. You schedule the application to be available in five days from today. This schedule gives you time to distribute the content and confirm its status. You then set the installation deadline for one month from today. Users see the application in Software Center when it's available in five days. If they do nothing, the client automatically installs the application at the installation deadline.

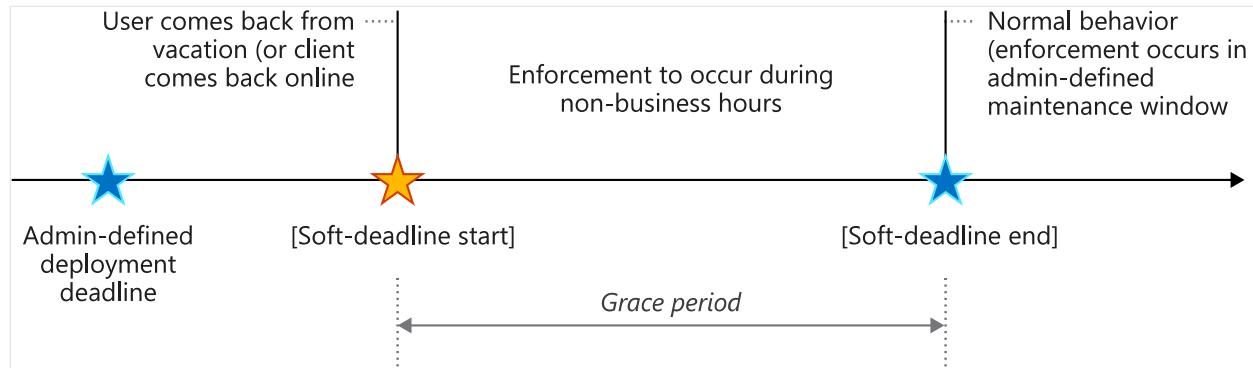
If the application you're deploying supersedes another application, set the installation deadline when users receive the new application. Set the **Installation Deadline** to upgrade users with the superseded application.

Delay enforcement with a grace period

You might want to give users more time to install required applications *beyond* any deadlines you set. This behavior is typically required when a computer is turned off for a long time, and needs to install many applications. For example, when a user returns from vacation, they have to wait for a long time as the client installs overdue deployments. To help solve this problem, define an enforcement grace period.

- First, configure this grace period with the property **Grace period for enforcement after deployment deadline (hours)** in client settings. For more information, see the [Computer agent](#) group. Specify a value between 1 and 120 hours.
- On the **Scheduling** page of a required application deployment, enable the option to **Delay enforcement of this deployment according to user preferences, up to the grace period defined in client settings**. The enforcement grace period applies to all deployments with this option enabled and targeted to devices to which you also deployed the client setting.

After the deadline, the client installs the application in the first non-business window, which the user configured, up to this grace period. However, the user can still open Software Center and install the application at any time. Once the grace period expires, enforcement reverts to normal behavior for overdue deployments.



① Note

Most of the time, this feature addresses the scenario when the device is powered off while the user is out of the office. Technically, the grace period starts when the client gets policy after the deployment deadline. The same behavior happens if you stop the Configuration Manager client service (CcmExec), and then restart it at some time after the deployment deadline.

User experience settings

On the **User Experience** page, specify information about how users can interact with the application installation.

- **User notifications**: Specify whether to display notification in Software Center at the configured available time. This setting also controls whether to notify users on the client computers. For available deployments, you can't select the option to **Hide in Software Center and all notifications**.
 - **When software changes are required, show a dialog window to the user instead of a toast notification**: Select this option to change the user experience to be more intrusive. It only applies to required deployments. For more information, see [User notifications](#).
- **Software Installation and System restart**: Only configure these settings for required deployments. They specify the behaviors when the deployment reaches the deadline outside of any defined maintenance windows. For more information about maintenance windows, see [How to use maintenance windows](#).

- **Write filter handling for Windows Embedded devices:** This setting controls the installation behavior on Windows Embedded devices that are enabled with a write filter. Choose the option to commit changes at the installation deadline or during a maintenance window. When you select this option, a restart is required and the changes persist on the device. Otherwise, the application is installed to the temporary overlay, and committed later.
 - When you deploy a software update to a Windows Embedded device, make sure the device is a member of a collection that has a configured maintenance window. For more information about maintenance windows and Windows Embedded devices, see [Create Windows Embedded applications](#).

Alerts

On the **Alerts** page, configure how Configuration Manager generates alerts for this deployment. If you're also using System Center Operations Manager, configure its alerts as well. You can only configure some alerts for required deployments.

Next steps

- [Monitor applications](#)
- [Disable and delete application deployments](#)
- [Troubleshoot application deployments](#)
- [Common error codes for app installation](#)
- [Management tasks for applications](#)
- [Software Center user guide](#)

Note

This article used to include more sections, which have moved to the following articles:

- [Delete a deployment](#)
- [User notifications for required deployments](#)
- [Check for running executable files](#)
- [Deploy user-available apps](#)

Create phased deployments with Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Phased deployments automate a coordinated, sequenced rollout of software across multiple collections. For example, deploy software to a pilot collection, and then automatically continue the rollout based on success criteria. Create phased deployments with the default of two phases, or manually configure multiple phases.

Create phased deployments for the following objects:

- **Task sequence**
 - The phased deployment of task sequences doesn't support PXE or media installation
- **Application**
- **Software update**
 - You can't use an automatic deployment rule (ADR) with a phased deployment

Prerequisites

Security scope

Deployments created by phased deployments aren't viewable to any administrative user that doesn't have the **All** security scope. For more information, see [Security scopes](#).

Distribute content

Before creating a phased deployment, distribute the associated content to a distribution point.

- **Application:** Select the target application in the console and use the **Distribute Content** action in the ribbon. For more information, see [Deploy and manage content](#).
- **Task sequence:** You have to create referenced objects like the OS upgrade package before creating the task sequence. Distribute these objects before creating a deployment. Use the **Distribute Content** action on each object, or the task sequence. To view status of all referenced content, select the task sequence, and

switch to the **References** tab in the details pane. For more information, see the specific object type in [Prepare for OS deployment](#).

- **Software update:** create the deployment package and distribute it. Use the Download Software Updates Wizard. For more information, see [Download software updates](#).

Phase settings

These settings are unique to phased deployments. Configure these settings when creating or editing the phases to control the scheduling and behavior of the phased deployment process.

Optionally, use the following Windows PowerShell cmdlets to manually configure phases for software update and task sequence phased deployments:

- [New-CMSoftwareUpdatePhase](#)
- [New-CMTaskSequencePhase](#)

Criteria for success of the first phase

- **Deployment success percentage:** Specify the percent of devices that need to successfully complete the deployment for the first phase to succeed. By default, this value is 95%. In other words, the site considers the first phase successful when the compliance state for 95% of the devices is **Success** for this deployment. The site then continues to the second phase, and creates a deployment of the software to the next collection.
- **Number of devices successfully deployed:** Specify the number of devices that need to successfully complete the deployment for the first phase to succeed. This option is useful when the size of the collection is variable, and you have a specific number of devices to show success before moving to the next phase.

Conditions for beginning second phase of deployment after success of the first phase

- **Automatically begin this phase after a deferral period (in days):** Choose the number of days to wait before beginning the second phase after the success of the first. By default, this value is one day.
- **Manually begin the second phase of deployment:** The site doesn't automatically begin the second phase after the first phase succeeds. This option requires that

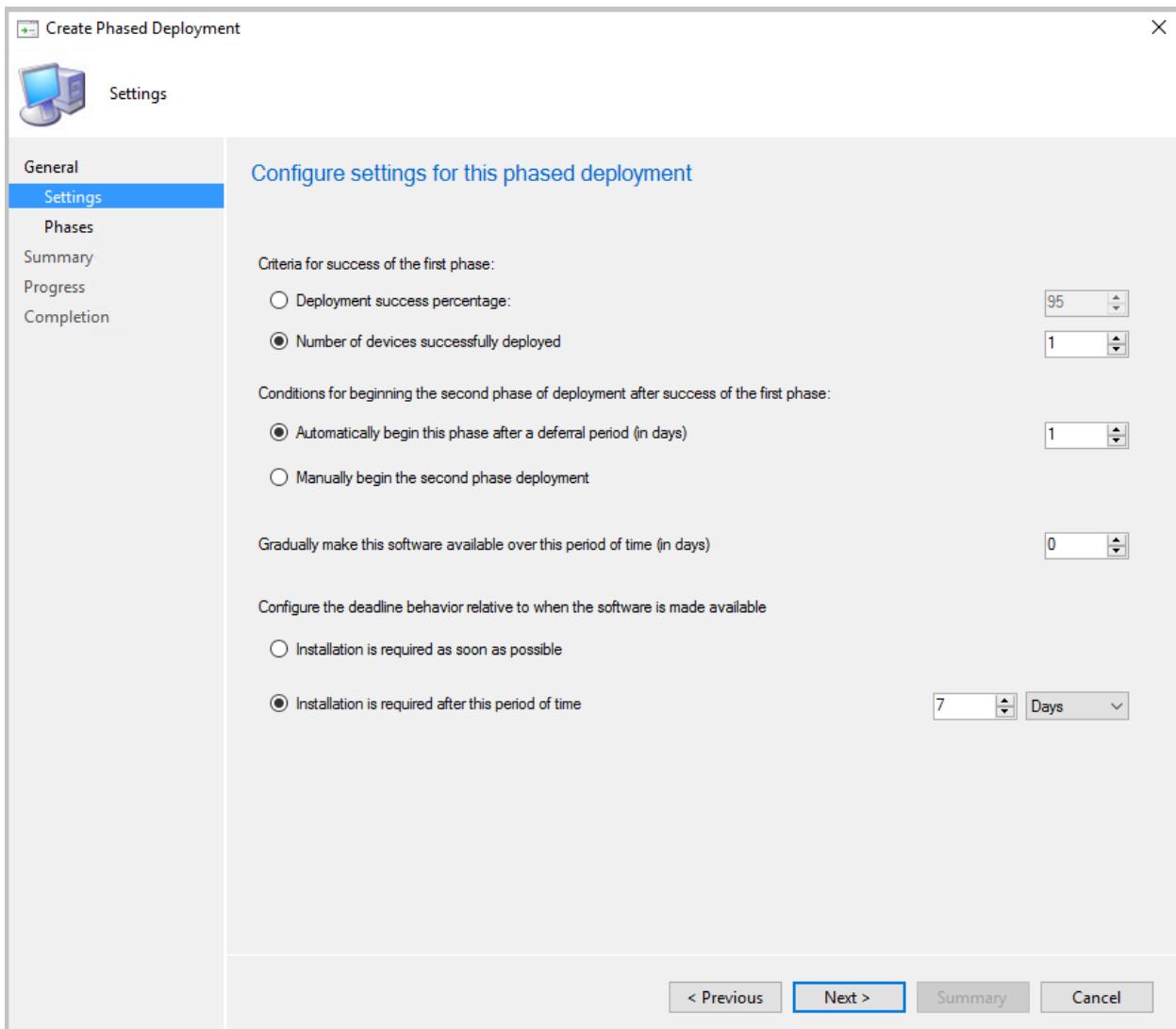
you manually start the second phase. For more information, see [Move to the next phase](#).

 **Note**

This option isn't available for phased deployments of applications.

Gradually make this software available over this period of time (in days)

Configure this setting for the rollout in each phase to happen gradually. This behavior helps mitigate the risk of deployment issues, and decreases the load on the network that is caused by the distribution of content to clients. The site gradually makes the software available depending on the configuration for each phase. Every client in a phase has a deadline relative to the time the software is made available. The time window between the available time and deadline is the same for all clients in a phase. The default value of this setting is zero, so by default the deployment isn't throttled. Don't set the value higher than 30.



Configure the deadline behavior relative to when the software is made available

- **Installation is required as soon as possible:** Set the deadline for installation on the device as soon as the device is targeted.
- **Installation is required after this period of time:** Set a deadline for installation a certain number of days after device is targeted. By default, this value is seven days.

Automatically create a default two-phase deployment

1. Start the Create Phased Deployment wizard in the Configuration Manager console.
This action varies based on the type of software you're deploying:
 - **Application:** Go to the Software Library, expand Application Management, and select Applications. Select an existing application, and then choose Create Phased Deployment in the ribbon.

- **Software update:** Go to the **Software Library**, expand **Software Updates**, and select **All Software Updates**. Select one or more updates, and then choose **Create Phased Deployment** in the ribbon.

This action is available for software updates from the following nodes:

- Software Updates
 - **All Software Updates**
 - **Software Update Groups**
- Windows Servicing, **All Windows Updates**
- Office 365 Client Management, **Office 365 Updates**
- **Task sequence:** Go to the **Software Library** workspace, expand **Operating Systems**, and select **Task Sequences**. Select an existing task sequence, and then choose **Create Phased Deployment** in the ribbon.

2. On the **General** page, give the phased deployment a **Name**, **Description** (optional), and select **Automatically create a default two phase deployment**.
3. Select **Browse** and choose a target collection for both the **First Collection** and **Second Collection** fields. For a task sequence and software updates, select from device collections. For an application, select from user or device collections. Select **Next**.

 **Important**

The Create Phased Deployment wizard doesn't notify you if a deployment is potentially high-risk. For more information, see [Settings to manage high-risk deployments](#) and the note when you [Deploy a task sequence](#).

4. On the **Settings** page, choose one option for each of the scheduling settings. For more information, see [Phase settings](#). Select **Next** when complete.
5. On the **Phases** page, see the two phases that the wizard creates for the specified collections. Select **Next**. These instructions cover the procedure to automatically create a default two-phase deployment. The wizard lets you add, remove, reorder, edit, or view phases for a phased deployment. For more information on these additional actions, see [Create a phased deployment with manually configured phases](#).
6. Confirm your selections on the **Summary** tab, and then select **Next** to complete the wizard.

 **Note**

Starting on April 21, 2020, Office 365 ProPlus is being renamed to **Microsoft 365 Apps for enterprise**. For more information, see [Name change for Office 365 ProPlus](#). You may still see the old name in the Configuration Manager product and documentation while the console is being updated.

Optionally, use the following Windows PowerShell cmdlets for this task:

- [New-CMApplicationAutoPhasedDeployment](#)
- [New-CMSoftwareUpdateAutoPhasedDeployment](#)
- [New-CMTaskSequenceAutoPhasedDeployment](#)

Create a phased deployment with manually configured phases

Create a phased deployment with manually configured phases for a task sequence. Add up to 10 additional phases from the **Phases** tab of the Create Phased Deployment wizard.

 **Note**

You can't currently manually create phases for an application. The wizard automatically creates two phases for application deployments.

1. Start the Create Phased Deployment wizard for either a task sequence or software updates.
2. On the **General** page of the Create Phased Deployment wizard, give the phased deployment a **Name**, **Description** (optional), and select **Manually configure all phases**.
3. From the **Phases** page of the Create Phased Deployment wizard, the following actions are available:
 - **Filter** the list of deployment phases. Enter a string of characters for a case-insensitive match of the Order, Name, or Collection columns.
 - **Add** a new phase:

- a. On the **General** page of the Add Phase Wizard, specify a **Name** for the phase, and then browse to the target **Phase Collection**. The additional settings on this page are the same as when normally deploying a task sequence or software updates.
- b. On the **Phase Settings** page of the Add Phase Wizard, configure the scheduling settings, and select **Next** when complete. For more information, see [Settings](#).

 **Note**

You can't edit the phase settings, **Deployment success percentage** or **Number of devices successfully deployed**, on the first phase. These settings only apply to phases that have a previous phase.

- c. The settings on the **User Experience** and **Distribution Points** pages of the Add Phase Wizard are the same as when normally deploying a task sequence or software updates.
- d. Review the settings on the **Summary** page, and then complete the Add Phase Wizard.
 - **Edit:** This action opens the selected phase's Properties window, which has tabs the same as the pages of the Add Phase Wizard.
 - **Remove:** This action deletes the selected phase.

 **Warning**

There is no confirmation, and no way to undo this action.

- **Move Up or Move Down:** The wizard orders the phases by how you add them. The most recently added phase is last in the list. To change the order, select a phase, and then use these buttons to move the phase's location in the list.

 **Important**

Review the phase settings after changing the order. Make sure the following settings are still consistent with your requirements for this phased deployment:

- Criteria for success of the previous phase

- Conditions for beginning this phase of deployment after success of the previous phase

4. Select **Next**. Review the settings on the **Summary** page, and then complete the Create Phased Deployment wizard.

Optionally, use the following Windows PowerShell cmdlets for this task:

- [New-CMSoftwareUpdateManualPhasedDeployment](#)
- [New-CMTaskSequenceManualPhasedDeployment](#)

After you create a phased deployment, open its properties to make changes:

- **Add** additional phases to an existing phased deployment.
- If a phase isn't active, you can **Edit**, **Remove**, or **Move** it up or down. You can't move it before an active phase.
- When a phase is active, it's read-only. You can't edit it, remove it, or move its location in the list. The only option is to **View** the properties of the phase.
- An application phased deployment is always read-only.

Next steps

Manage and monitor phased deployments:

- [Application](#)
- [Software update](#)
- [Task sequence](#)

Approve applications in Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

When [deploying an application](#) in Configuration Manager, you can require approval before installation. Users request the application in Software Center, and then you review the request in the Configuration Manager console. You can approve or deny the request.

ⓘ Note

Starting in version 2111, you can also use most approval behaviors with [application groups](#).

Approval settings

The application approval behavior depends upon whether you enable the recommended [optional app approval experience](#). One of the following approval settings appears on the [Deployment Settings](#) page of the application deployment:

An administrator must approve a request for this application on the device

ⓘ Note

Configuration Manager doesn't enable this feature by default. Before using it, enable the optional feature **Approve application requests for users per device**. For more information, see [Enable optional features from updates](#).

If you don't enable this feature, you see the [prior experience](#).

The administrator approves any user requests for the application before the user can install it on the requested device. If the administrator approves the request, the user is only able to install the application on that device. The user must submit another request to install the application on another device. This option is grayed out when the

deployment purpose is **Required**, or when you deploy the application to a device collection.

Note

To take advantage of new Configuration Manager features, first update clients to the latest version. While new functionality appears in the Configuration Manager console when you update the site and console, the complete scenario isn't functional until the client version is also the latest.

View Application Requests under Application Management in the **Software Library** workspace of the Configuration Manager console. There's a **Device** column in the list for each request. When you take action on the request, the Application Request dialog also includes the device name from which the user submitted the request.

If a request isn't approved within 30 days, it's removed. Reinstalling the client might cancel any pending approval requests.

When you require approval on a deployment to a device collection, the app isn't displayed in Software Center. If you require approval on a deployment to a user collection, the app is displayed in Software Center. You can still hide it from users with the client setting, **Hide unapproved applications in Software Center**. For more information, see [Software Center client settings](#).

After you've approved an application for installation, you can **Deny** the request in the Configuration Manager console. If users haven't already installed the application, this action stops them from installing new copies of the application from Software Center. If an application was previously approved and installed, when you **Deny** the request for the application, the client uninstalls the application from the user's device.

If you approve an app request in the console, and then deny it, you can approve it again. The app is reinstalled on the client after you approve it.

Automate the approval process with the [Approve-CMAApprovalRequest](#) PowerShell cmdlet. This cmdlet includes the **InstallActionBehavior** parameter. Use this parameter to specify whether to install the application right away or during non-business hours.

You can see which deployments require approval. Select an app in the **Applications** node. In the details pane, switch to the **Deployments** tab. There's a column displayed by default, **Requires Approval**.

Retry the install of pre-approved applications

You can retry the installation of an app that you previously approved for a user or device. The approval option is only for available deployments. If the user uninstalls the app, or if the initial install process fails, Configuration Manager doesn't reevaluate its state and reinstall it. This feature allows a support technician to quickly retry the app install for a user that calls for help.

1. Open the Configuration Manager console as a user that has the **Approve** permission on the Application object. For example, the **Application Administrator** or **Application Author** built-in roles have this permission.
2. Deploy an app that requires approval, and approve it.

 **Tip**

Alternatively, **install an application for a device**. It creates an approved request for the app on the device.

If the application doesn't install successfully, or the user uninstalls the app, use the following process to retry:

1. In the Configuration Manager console, go to the **Software Library** workspace, expand **Application Management**, and select the **Application Requests** node.
2. Select the previously approved app. In the Approval Request group of the ribbon, select **Retry install**.

Other app approval resources

- [Application approval improvements in ConfigMgr 1810](#)
- [Updates to the application approval process in Configuration Manager](#)

Require administrator approval if users request this application

 **Note**

This experience applies if you don't enable the recommended [optional app approval experience](#).

The administrator approves any user requests for the application before the user can install it. This option is grayed out when the deployment purpose is **Required**, or when

you deploy the application to a device collection.

Application approval requests are displayed in the **Application Requests** node, under **Application Management** in the **Software Library** workspace. If a request isn't approved within 30 days, it's removed. Reinstalling the client might cancel any pending approval requests.

After you've approved an application for installation, you can **Deny** the request in the Configuration Manager console. This action doesn't cause the client to uninstall the application from any devices. It stops users from installing new copies of the application from Software Center.

Email notifications

You can configure email notifications for application approval requests. When a user requests an application, you receive an email. Click links in the email to approve or deny the request, without requiring the Configuration Manager console.

You can define the email addresses of the users who can approve or deny the request while creating a new deployment for the application. If you need to change the list of email addresses afterwards, go to the **Monitoring** workspace, expand **Alerts**, and select the **Subscriptions** node. Select **Properties** from one of the **Approve application via email** subscriptions that's related to your application deployment.

If there is more than one alert, you can determine which alert goes with which deployment. Open the alert properties, and view the list of **Selected alerts** on the General tab. The deployment is enabled as the alert for this subscription.

Users can add a comment to the request from Software Center. This comment shows on the application request in the Configuration Manager console. That comment also shows in the email. Including this comment in the email helps the approvers make a better decision to approve or deny the request.

Prerequisites

To send email notifications and take action on internal network

With these prerequisites, recipients receive an email with notification of the request. If they are on the internal network, they can also approve or deny the request from the email.

- Enable the optional feature [Approve application requests for users per device](#).

- Configure [email notification for alerts](#).

 **Note**

The administrative user that deploys the application needs permission to create an alert and subscription. If this user doesn't have these permissions, they'll see an error at the end of the **Deploy Software Wizard**: "You do not have security rights to perform this operation."

- Set up the administration service in Configuration Manager.

 **Note**

If you have multiple child primary sites in a hierarchy, configure these prerequisites for each primary site where you want to enable this feature. The links in the email notification are for the administration service at the primary site.

To take action from internet

With these additional optional prerequisites, recipients can approve or deny the request from anywhere they have internet access.

- Enable the SMS Provider administration service through the cloud management gateway. In the Configuration Manager console, go to the **Administration** workspace, expand **Site Configuration**, and select the **Servers and Site System Roles** node. Select the server with the SMS Provider role. In the details pane, select the **SMS Provider** role, and select **Properties** in the ribbon on the Site Role tab. Select the option to **Allow Configuration Manager cloud management gateway traffic for administration service**.
- Install a supported version of the .NET Framework. Starting in version 2107, the SMS Provider requires .NET version 4.6.2, and version 4.8 is recommended. In version 2103 and earlier, this role requires .NET 4.5 or later. For more information, [Site and site system prerequisites](#).
- Set up a [cloud management gateway](#).

 **Note**

This scenario doesn't support CMG deployments with a virtual machine scale set until Configuration Manager version 2207 or later is installed.

- Onboard the site to [Azure services](#) for Cloud Management.
- Enable [Azure AD User Discovery](#).
- Manually configure settings in Azure AD:
 1. Go to the [Azure portal](#) ↗ as a user with *Global Admin* permissions. Go to **Azure Active Directory**, and select **App registrations**.
 2. Select the application that you created for Configuration Manager **Cloud Management** integration.
 3. In the **Manage** menu, select **Authentication**.
 - a. In the **Redirect URIs** section, paste in the following path: `https://<CMG FQDN>/CCM_Proxy_ServerAuth/ImplicitAuth`
 - b. Replace `<CMG FQDN>` with the fully qualified domain name (FQDN) of your cloud management gateway (CMG) service. For example, `GraniteFalls.Contoso.com`.
 - c. For Configuration Manager version 2111 and later, in the **Implicit grant and hybrid flows** section, select the following options:
 - **Access tokens (used for implicit flows)**
 - **ID tokens (used for implicit and hybrid flows)**
 - d. Then select **Save**.
 4. For Configuration Manager version 2107 and earlier, in the **Manage** menu, select **Manifest**.
 - a. In the Edit manifest pane, find the **oauth2AllowImplicitFlow** property.
 - b. Change its value to **true**. For example, the entire line should look like the following line: `"oauth2AllowImplicitFlow": true,`
 - c. Select **Save**.

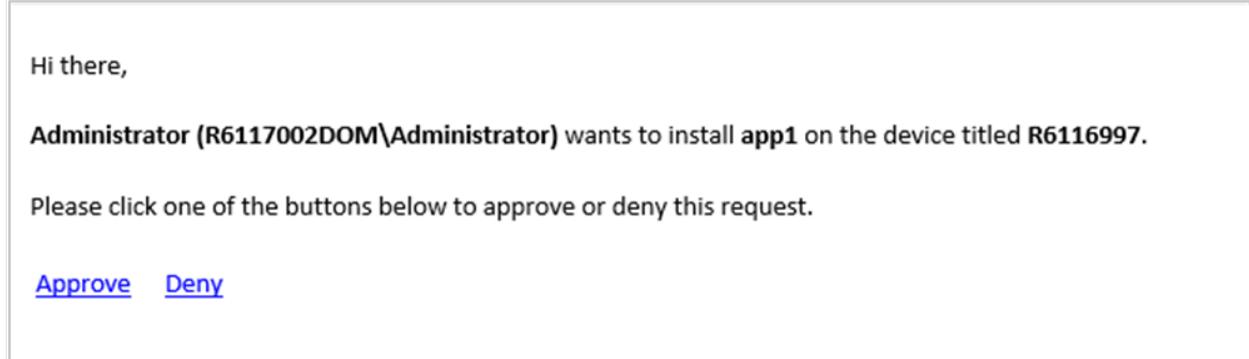
Configure email approval

1. In the Configuration Manager console, [deploy an application](#) as available to a user collection. On the **Deployment Settings** page, enable it for approval. Then enter one or more email addresses to receive notification. Separate email addresses with a semi-colon (;).

 **Note**

Anyone in your Azure AD organization who receives the email can approve the request. Don't forward the email to others unless you want them to take action.

2. As a user, request the application in Software Center.
3. You receive an email notification within five minutes. The content of the email is similar to the following example:



Hi there,

Administrator (R6117002DOM\Administrator) wants to install **app1** on the device titled **R6116997**.

Please click one of the buttons below to approve or deny this request.

[Approve](#) [Deny](#)

 **Note**

The link to approve or deny is for one-time use. For example, you configure a group alias to receive notifications. Meg approves the request. Now Bruce can't deny the request.

Review the **NotiCtrl.log** file on the site server for troubleshooting.

Maintenance

Configuration Manager stores the information about the application approval request in the site database. For requests that are canceled or denied, the site deletes the request history after 30 days. You can configure this deletion behavior with the **Delete Aged Application Request Data** site maintenance task. The site never deletes any approved or pending application requests.

Next steps

[Monitor applications from the Configuration Manager console](#)

Install applications for a device

Article • 10/04/2022

From the Configuration Manager console you can install applications to a device in real time. This feature can help reduce the need for separate collections for every application.

① Note

Starting in version 2111, this behavior also supports **application groups**. When this article refers to an *application*, it also applies to app groups.

Prerequisites

- Enable the [optional feature](#) **Approve application requests for users per device**.
- [Deploy the application](#) as *Available* to a device collection.
 - On the **Deployment Settings** page of the deployment wizard, select the following option: **An administrator must approve a request for this application on the device**.

① Note

With these deployment settings, no policy is sent to the client. The app isn't shown as available in Software Center, and a user can't install the app with this deployment. After you use this action to install the app, the user can run it, and see its installation status in Software Center.

- Your user account needs the following permissions:
 - **Application:** Read, Approve
 - **Collection:** Read, Read Resource, Modify Resource, View Collected File

For example, the **Application Administrator** built-in role has these permissions.

💡 Tip

In a hierarchy, wait for application and deployment information to replicate to the primary site to which the target client is assigned.

Process

1. In the Configuration Manager console, go to the **Assets and Compliance** workspace, and select the **Devices** node. Select the target device, and then select the **Install application** action in the ribbon. Starting in version 2111, select the **Install Application Group** action for an app group.
2. Select one or more applications from the list. The list only shows applications that you already deployed with the prerequisite settings.

This action triggers the installation of the selected pre-deployed applications on the device.

To see status of the approval request, in the **Software Library** workspace, expand **Application Management**, and select the **Application Requests** node.

Monitor the app installation the same as usual in the **Deployments** node of the **Monitoring** workspace.

See also

[Approve applications](#)

Check for running executable files

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Configure an application deployment to check if certain executable files are running on the client. Use this option to check for processes that might disrupt the installation of the application. If one of these executable files is running, the client blocks the installation of the deployment type. The user must close the running executable file before the client can install the deployment type. For deployments with a purpose of required, the client can automatically close the running executable file.

1. Open the **Properties** for the deployment type.
2. Switch to the **Install Behavior** tab, and select **Add**.
3. In the **Add Executable File** window, enter the name of the target executable file. Optionally, enter a friendly name for the application to help you identify it in the list.
4. Select **OK** to save and close the deployment type properties window.
5. When you deploy the application, select the option to **Automatically close any running executables you specified on the install behavior tab of the deployment type properties dialog box**. This option is on the **Deployment Settings** tab of the deployment properties.

ⓘ Note

If you configure an application to check for running executable files, and include it in the **Install Application** task sequence step, the task sequence will fail to install it. If you don't configure this task sequence step to continue on error, then the entire task sequence fails.

Client behaviors and user notifications

After clients receive the deployment, the following behavior applies:

- If you deployed the application as **Available**, and a user tries to install it, the client prompts the user to close the specified running executable files before proceeding with the installation.

- If you deployed the application as **Required**, and specified to **Automatically close any running executables you specified on the install behavior tab of the deployment type properties dialog box**, then the client displays a notification. It informs the user that the specified executable files are automatically closed when the application installation deadline is reached. If the user tries to install the application before the deadline, the deployment will fail. It notifies the user that the installation couldn't complete because the specified executables are running.
 - Schedule these dialogs in the **Computer Agent** group of client settings. For more information, see [Computer agent](#).
 - If you don't want the user to see these messages, select the option to **Hide in Software Center and all notifications** on the **User Experience** tab of the deployment's properties. For more information, see [User Experience settings](#).
- If you deployed the application as **Required**, and didn't specify to **Automatically close any running executables you specified on the install behavior tab of the deployment type properties dialog box**, then the installation of the app fails if one or more of the specified applications are running.

Next steps

- Plan for [user notifications](#) when you deploy applications
- [Create deployment types for an application](#)
- [Deploy applications](#)

Share an application from Software Center

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

You can copy a hyperlink to an application in Software Center using the  Share button in the Application Details view. You can only share hyperlinks for applications. If the application becomes unavailable, the hyperlink opens a window with an application unavailable message.

1. Choose **Applications**, and then choose the application.
2. Select the  Share button.
3. Select **Copy** in the window.
4. Paste the URL into an email to share the application.

Tip

To create a link in an Outlook email, press **CTRL + K** and then paste the URL.

Simulate application deployments with Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

You can use simulated deployments to test an application deployment without installing or uninstalling the application. A simulated deployment evaluates the detection method, requirements, and dependencies for a deployment type. It reports the results in the **Deployments** node of the **Monitoring** workspace. Use the procedure in this topic to simulate an application deployment in Configuration Manager.

Note

You cannot use simulated deployments for collections of mobile devices.

You cannot deploy an application with a deployment purpose of **Uninstall** if a simulated deployment of the same application is active.

Configure a simulated application deployment

1. In the Configuration Manager console, select one of the following:
 - A collection of users.
 - A collection of devices.
 - A Configuration Manager application.
2. On the **Home** tab, in the **Deployment** group, choose **Simulate Deployment**.
3. In the Simulate Application Deployment Wizard, set the following details for your simulated deployment:
 - **Application.** Choose **Browse**, and then select the application you want to create a simulated deployment for.
 - **Collection.** Choose **Browse**, and then select the collection that you want to use for the simulated deployment.
 - **Action.** From the drop-down list, select whether you want to simulate the installation or the uninstallation of the selected application.

- **Deploy automatically with or without user login.** If this option is checked, the clients evaluate the simulated deployment whether or not the clients are logged in.

4. Click **Next**, review the information on the **Summary** page, and then finish the wizard to create the simulated application deployment.

5. Simulated applications appear in the **Deployments** node of the **Monitoring** workspace, with a purpose of **Simulate**. For more information about how to monitor application deployments, see [Monitor applications from the Configuration Manager console](#).

Microsoft Edge Management

Article • 10/04/2022

Applies to: Configuration Manager (Current Branch)

The all-new Microsoft Edge is ready for business. You can deploy [Microsoft Edge, version 77 and later](#) to your users. A PowerShell script is used to install the Microsoft Edge build selected. The script also turns off automatic updates for Microsoft Edge so they can be managed with Configuration Manager.

Deploy Microsoft Edge

Admins can pick the Beta, Dev, or Stable channel, along with a version of the Microsoft Edge client to deploy. Each release incorporates learnings and improvements from our customers and community. For more information, see [Microsoft Edge release schedule](#).

Prerequisites for deploying

For clients targeted with a Microsoft Edge deployment:

- PowerShell [Execution Policy](#) can't be set to Restricted.
 - PowerShell is executed to perform the installation.
- The Microsoft Edge installer, Attack Surface Reduction rules engine for tenant attach, and [CMPivot](#) are currently signed with the [Microsoft Code Signing PCA 2011](#) certificate. If you set PowerShell execution policy to [AllSigned](#), then you need to make sure that devices trust this signing certificate. You can export the certificate from a computer where you've installed the Configuration Manager console. View the certificate on `"C:\Program Files (x86)\Microsoft Endpoint Manager\AdminConsole\bin\CMSpivot.exe"`, and then export the code signing certificate from the certification path. Then import it to the *machine's Trusted Publishers* store on managed devices. You can use the process in the following blog, but make sure to export the *code signing certificate* from the certification path: [Adding a Certificate to Trusted Publishers using Intune](#).

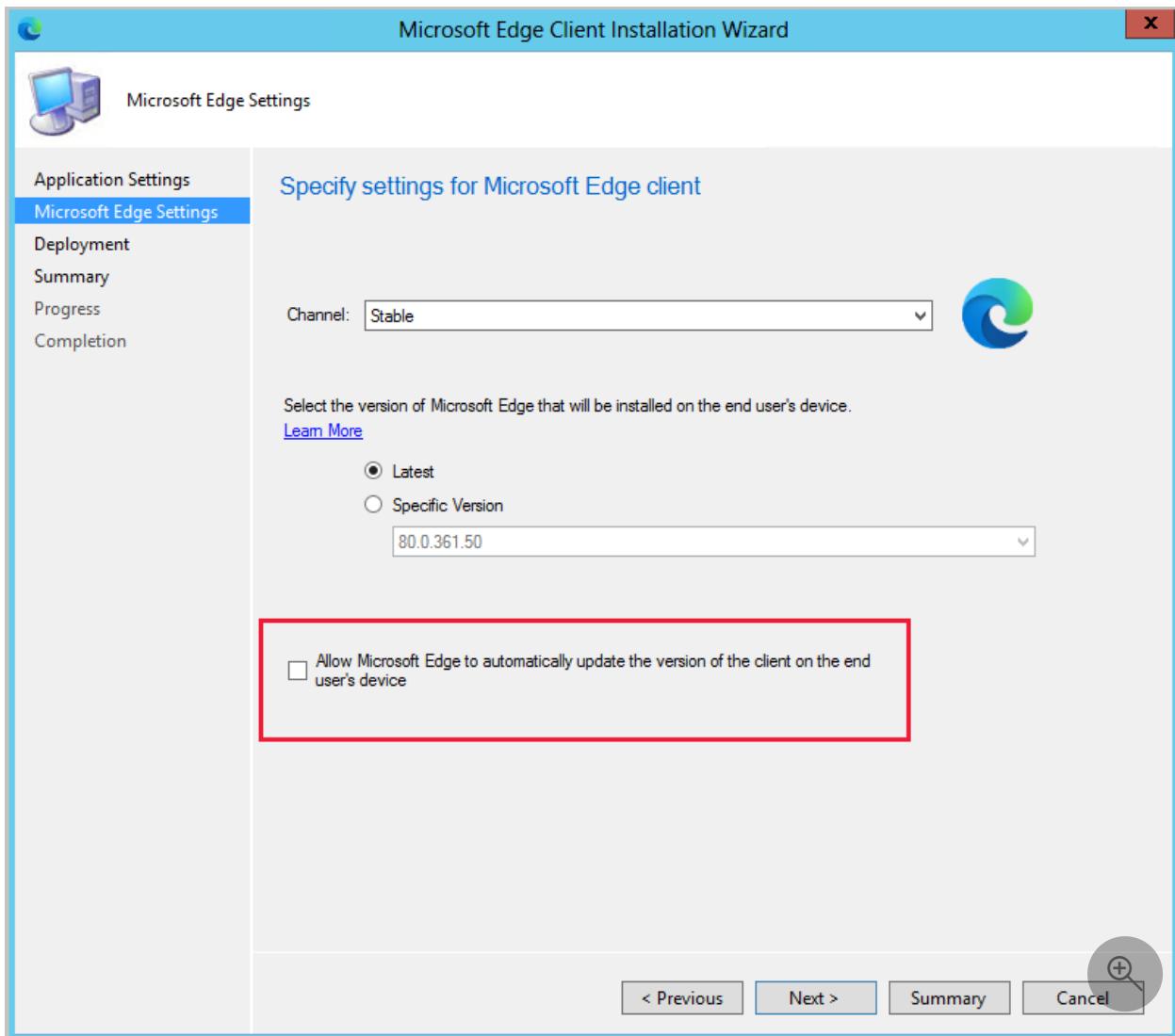
The device running the Configuration Manager console needs access to the following endpoints for deploying Microsoft Edge:

Location	Use
----------	-----

Location	Use
<code>https://aka.ms/cmedgeapi</code>	Information about releases of Microsoft Edge
<code>https://edgeupdates.microsoft.com/api/products?view=enterprise</code>	Information about releases of Microsoft Edge
<code>http://dl.delivery.mp.microsoft.com</code>	Content for Microsoft Edge releases

Verify Microsoft Edge update policies

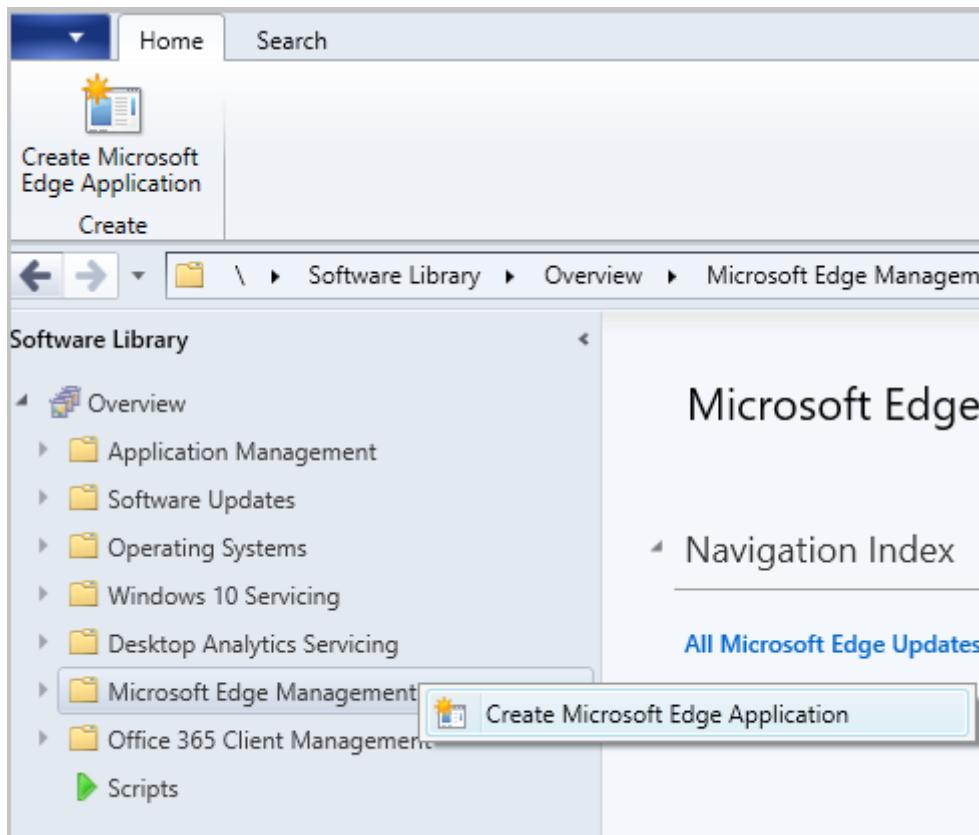
Starting in version 2002, you can create a Microsoft Edge application that's set up to receive automatic updates rather than having automatic updates disabled. This change allows you to choose to manage updates for Microsoft Edge with Configuration Manager or allow Microsoft Edge to automatically update. When creating the application, select **Allow Microsoft Edge to automatically update the version of the client on the end user's device** on the **Microsoft Edge Settings** page. If you previously used Group Policy to change this behavior, Group Policy will overwrite the setting made by Configuration Manager during installation of Microsoft Edge. For more information, see [Microsoft Edge update policies](#).



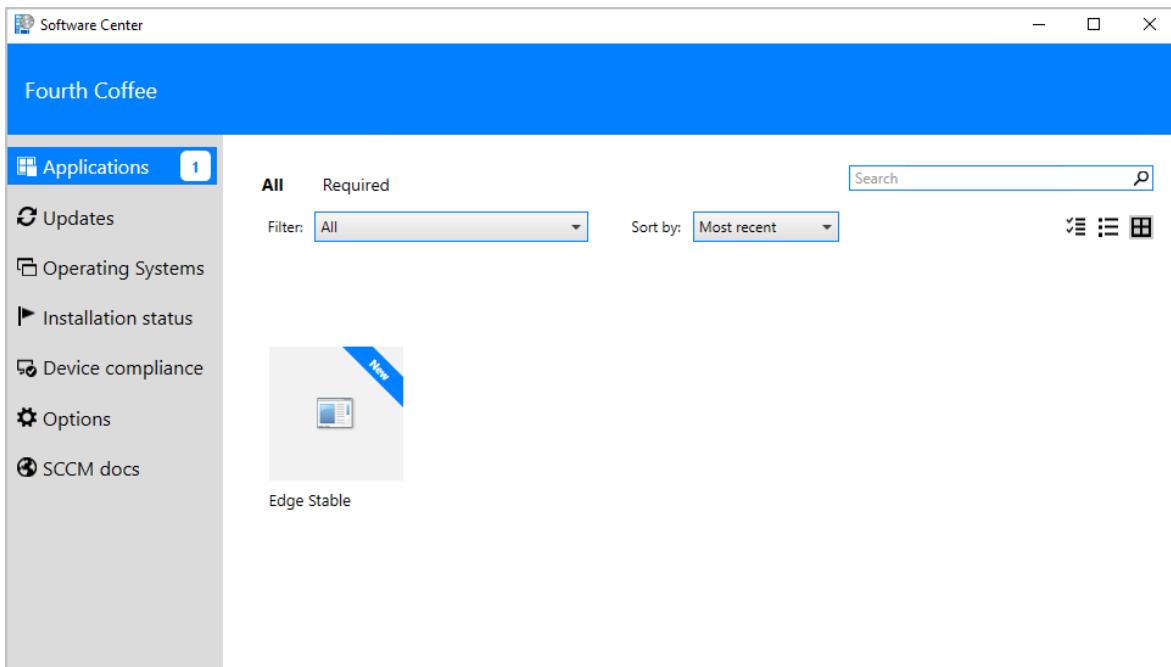
Create a deployment

Create a Microsoft Edge application using the built-in application experience, which makes Microsoft Edge easier to manage:

1. In the console, under **Software Library**, there's a new node called **Microsoft Edge Management**.
2. Select **Create Microsoft Edge Application** from either the ribbon, or by right-clicking on the **Microsoft Edge Management** node.



3. On the **Application Settings** page of the wizard, specify a name, description, and location for the content for the app. Ensure the content location folder you specify is empty.
4. On the **Microsoft Edge Settings** page, select:
 - The channel to deploy
 - The version to deploy
 - If you want to **Allow Microsoft Edge to automatically update the version of the client on the end user's device** (added in version 2002)
5. On the **Deployment** page, decide if you want to deploy the application. If you select **Yes**, you can specify your deployment settings for the application. For more information about deployment settings, see [Deploy applications](#).
6. In **Software Center** on the client device, the user can see and install the application.



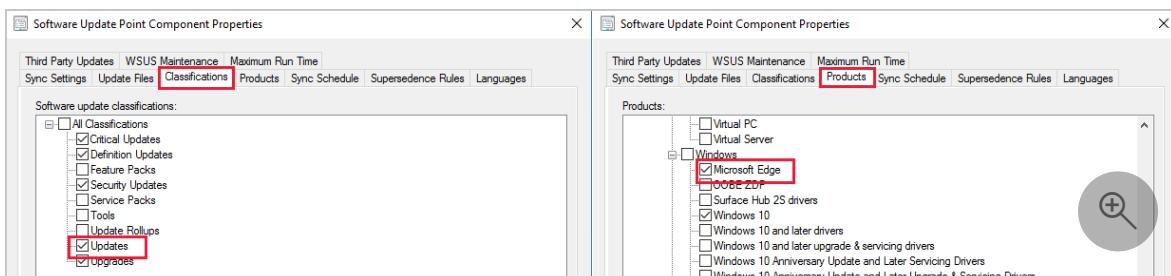
Log files for deployment

Location	Log	Use
Site server	SMSProv.log	Shows details if the creation of the app or deployment fails.
Varies	PatchDownloader.log	Shows details if the content download fails
Client	AppEnforce.log	Shows installation information

Update Microsoft Edge

The **All Microsoft Edge updates** node is under **Microsoft Edge Management**. This node helps you manage updates for all Microsoft Edge channels.

1. To get updates for Microsoft Edge, ensure you have the **Updates** classification and the **Microsoft Edge** product [selected for synchronization](#).



2. In the **Software Library** workspace, expand **Microsoft Edge Management** and click on the **All Microsoft Edge Updates** node.

3. If needed, click **Synchronize Software Updates** in the ribbon to start a synchronization. For more information, see [Synchronize software updates](#).

Icon	Title	Date Released or Revised
	Microsoft Edge-Dev Channel Version 81 Update for x64 based Editions (Build 81.0.389.2)	1/15/2020 11:19 AM
	Microsoft Edge-Dev Channel Version 81 Update for x86 based Editions (Build 81.0.389.2)	1/15/2020 11:19 AM
	Microsoft Edge-Beta Channel Version 79 Update for x86 based Editions (Build 79.0.309.65)	1/15/2020 10:07 AM
	Microsoft Edge-Beta Channel Version 79 Update for x64 based Editions (Build 79.0.309.65)	1/15/2020 10:07 AM
	Microsoft Edge-Stable Channel Version 79 Update for x86 based Editions (Build 79.0.309.65)	1/15/2020 10:04 AM
	Microsoft Edge-Stable Channel Version 79 Update for x64 based Editions (Build 79.0.309.65)	1/15/2020 10:04 AM

4. Manage and deploy Microsoft Edge updates like any other update, such as adding them to your [automatic deployment rule](#). Some of the common updates tasks you can do from the **All Microsoft Edge Updates** node include:

- [Create a phased deployment](#)
- [Manually deploy software updates](#)
- [Download software updates](#)

Microsoft Edge Management dashboard

Starting in Configuration Manager 2002, the Microsoft Edge Management dashboard provides you insights on the usage of Microsoft Edge and other browsers. In this dashboard, you can:

- See how many of your devices have Microsoft Edge installed
- See how many clients have different versions of Microsoft Edge installed.
 - This chart doesn't include Canary Channel.
- Have a view of the installed browsers across devices
- Have a view of preferred browser by device
 - Currently for the 2002 release, this chart will be empty.

Prerequisites for the dashboard

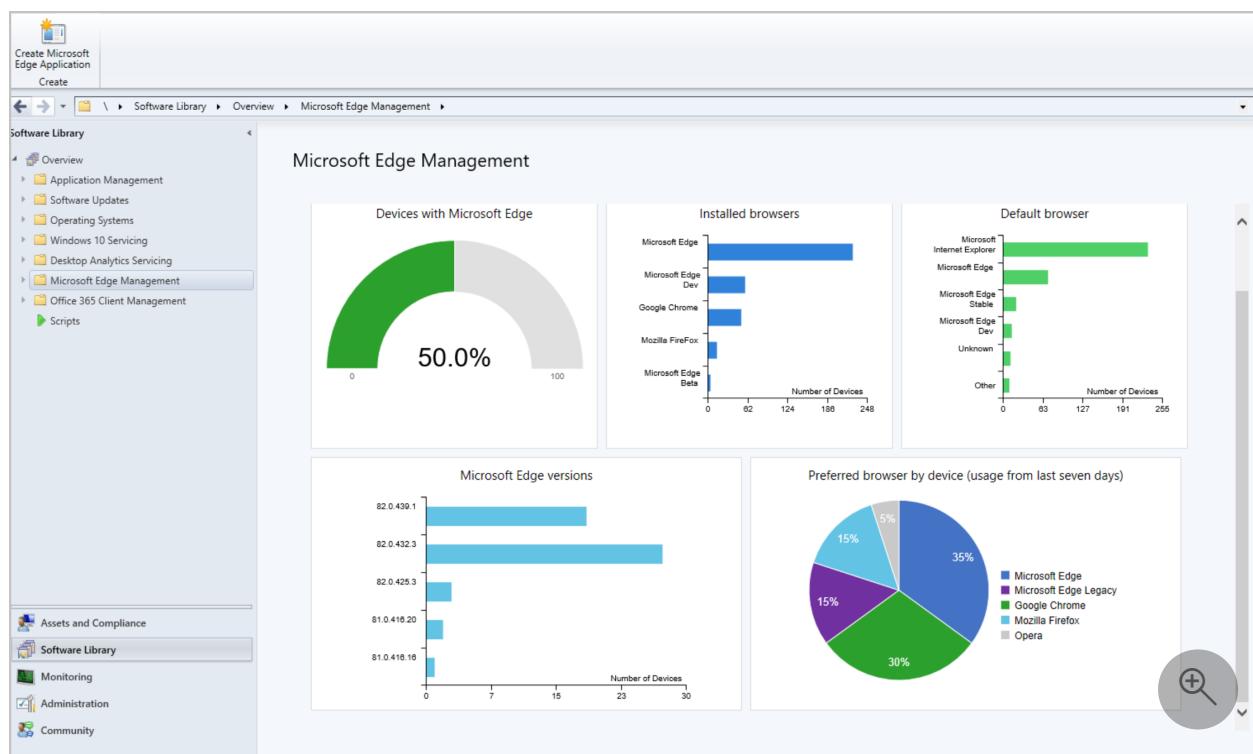
For Configuration Manager version 2203 or later, the [WebView2 console extension](#) must be installed. If needed, select the notification bell in the top right corner of the console to install the extension.

Enable the following properties in the below [hardware inventory](#) classes for the Microsoft Edge Management dashboard:

- **Installed Software - Asset Intelligence (SMS_InstalledSoftware)**
 - Software Code
 - Product Name
 - Product Version
- **Default Browser (SMS_DefaultBrowser)**
 - Browser Program ID
- **Browser Usage (SMS_BrowserUsage)**
 - BrowserName
 - UsagePercentage

View the dashboard

From the **Software Library** workspace, click **Microsoft Edge Management** to see the dashboard. Change the collection for the graph data by clicking **Browse** and choosing another collection. By default your five largest collections are in the drop-down list. When you select a collection that isn't in the list, the newly selected collection takes the bottom spot on your drop-down list.



Tip

The **Power BI sample reports** for Configuration Manager includes a report called **Edge Status**. This report can also help with monitoring Edge deployment.

Known issues

Hardware inventory may fail to process

Hardware inventory for devices might fail to process. Errors similar to the one below may be seen in the Dataldr.log file:

text

```
Begin transaction: Machine=<machine>
*** [23000][2627][Microsoft][SQL Server Native Client 11.0][SQL
Server]Violation of PRIMARY KEY constraint 'BROWSER_USAGE_HIST_PK'. Cannot
insert duplicate key in object 'dbo.BROWSER_USAGE_HIST'. The duplicate key
value is (XXXX, Y). : dbo.dBROWSER_USAGE_DATA
ERROR - SQL Error in
ERROR - is NOT retryable.
Rollback transaction: XXXX
```

Mitigation: To work around this issue, disable the collection of the Browser Usage (SMS_BrowserUsage) hardware inventory class.

Next steps

[Monitor applications](#)

[Monitor software updates](#)

[Manage and monitor phased deployments](#)

Deploy App-V virtual applications with Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

When you use Configuration Manager to manage virtual applications, you gain the following benefits:

- A single management infrastructure
- Scalability, deployment, and content distribution features, like collections and user device affinity
- Advanced application management features
- Operating system deployment, software and hardware inventory, software metering, and asset intelligence to support virtual applications

For more information about how to create and sequence applications with Microsoft Application Virtualization (App-V), see [Application Virtualization 4 documentation](#).

In addition to the other Configuration Manager requirements and procedures for creating an application, you must take the following considerations into account when you create and deploy virtual applications:

- To deploy virtual applications to computers, you must have the Configuration Manager client and App-V Client installed on your computers. Client devices can include desktop and portable computers, and Virtual Desktop Infrastructure (VDI) clients. The Configuration Manager and App-V Client software work together to deliver, locate, and launch virtual application packages. The Configuration Manager client manages the delivery of virtual application packages to the App-V Client. The App-V Client runs the virtual application on the client.
- To deploy a virtual application, you must first create the virtual application by using the App-V Application Virtualization Sequencer. The sequencer monitors the installation and setup process for an application and records the information that is needed for the application to run in a virtual environment. You can also use the sequencer to set which files and configurations apply to all users, and which configurations users can customize.
- When you sequence an application, you must save the package to a location that Configuration Manager can access. You can then create an application deployment

that contains this virtual application.

- Configuration Manager does not support the use of the shared read-only cache feature of App-V 4.6.
- Configuration Manager supports the Shared Content Store feature in App-V 5.
- When you create a deployment type for a virtual application, Configuration Manager creates the deployment type by using the contents of the application manifest file. This is an XML file that has information about the virtual application. Additionally, Configuration Manager creates requirements for the deployment type based on the contents of the App-V .osd file that has information about the supported operating systems for the virtual application.
- To deploy virtual applications in Configuration Manager, client computers must have at minimum the App-V 4.6 SP1 or a later version of the client installed.
- Before you can successfully deploy virtual applications, update the App-V client with the latest hotfix.
- When you use connection groups in App-V 5.0, your deployed virtual applications can share the same file system and registry on client computers. Unlike standard virtual applications, these applications can share data with one another. Additionally, connection groups preserve user settings for the applications that they contain. App-V virtual environments in Configuration Manager are used to set up connection groups on client computers. Virtual environments are created or changed on client computers when the application is installed or when clients next evaluate their installed applications. You can prioritize these applications so that when multiple applications try to change a file system or registry value, the application that has the highest priority takes precedence. For more information, see [Create App-V virtual environments](#).

Supported App-V versions

Configuration Manager supports the following versions of App-V:

- **App-V 4.6:** To use virtual applications in Configuration Manager, client computers must have the App-V 4.6 SP1, App-V 4.6 SP2, or App-V 4.6 SP3 client installed.
Before you can successfully deploy virtual applications, update the App-V 4.6 client with the latest hotfix.
- **App-V 5, App-V 5.0 SP1, App-V 5.0 SP2, App-V 5.0 SP3, and App-V 5.1:** For App-V 5.0 SP2, you must install [Hotfix Package 5](#) or use App-V 5.0 SP3.

- **App-V 5.2:** This is built into Windows 10 Education (1607 and later), Windows 10 Enterprise (1607 and later), and Windows Server 2016.

For more information about App-V in Windows 10, see the following topics:

- [What's new in App-V](#)
- [Getting Started with App-V for Windows 10](#)
- [Upgrading to App-V for Windows 10 from an existing installation](#)

Steps to manage App-V virtual applications

To manage App-V virtual applications, follow these steps:

1. **Sequence:** Sequencing is the process of converting an application into a virtual application by using the App-V sequencer.
2. **Create:** Use the Create Deployment Type Wizard to import the sequenced application into a Configuration Manager deployment type that you can then add to an application. You can also create virtual environments that allow multiple virtual applications to share settings.
3. **Distribute:** Distribution is the process of making App-V applications available on Configuration Manager distribution points.
4. **Deploy:** Deployment is the process of making the application available on client computers. This is called publishing and streaming in an App-V full infrastructure.

Configuration Manager virtual application delivery methods

Configuration Manager supports two methods for delivery of virtual applications to clients: streaming delivery and local delivery (download and execute).

When you're deciding which delivery method to use, compare the reduced disk space requirement for streaming delivery against the guaranteed availability of App-V applications in local delivery. The increased client disk space that is required for local delivery might be preferable to streaming delivery so that users always have the application available from any location.

Streaming delivery

When you use Configuration Manager to manage the App-V Client, it supports the streaming of virtual applications through HTTP or HTTPS from a distribution point. Streaming through HTTP or HTTPS is enabled by default and is set up in the dialog box for distribution point properties. When you deploy a virtual application to client computers and a user runs the virtual application, the Configuration Manager client contacts a management point to determine which distribution point to use. Then, the application is streamed from the distribution point.

Use the information in this table to help you decide if streaming delivery is the best delivery method for you:

Advantages	Disadvantages
-------------------	----------------------

Advantages	Disadvantages
This method uses standard network protocols to stream package content from distribution points.	Virtual applications are not streamed until the user runs the application for the first time. In this scenario, a user might receive program shortcuts for virtual applications and then disconnect from the network before running the virtual applications for the first time. If the user tries to run the virtual application while the client is offline, the user sees an error and can't run the virtualized application
Program shortcuts for virtual applications invoke a connection to the distribution point, so the virtual application delivery is on demand.	because a Configuration Manager distribution point is not available to stream the application. The application will be unavailable until the user reconnects to the network and runs the application.
To avoid this, you can use the local delivery method for virtual application delivery to clients, or you can enable the Internet-based client management for streaming delivery.	
This method works well for clients with high-bandwidth connections to the distribution points.	
Updated virtual applications distributed throughout the enterprise are available as clients receive policy that informs them that the current version is superseded and they download only the changes from the previous version.	
Access permissions are defined at the distribution point to prevent users from accessing unauthorized applications or packages.	

Local delivery (download and execute)

Download and execute is most common approach when using Configuration Manager because this approach closely mimics how other application formats are delivered with Configuration Manager. When you use the local delivery method, the Configuration Manager client first downloads the entire virtual application package into the

Configuration Manager client cache. The Configuration Manager then instructs the App-V Client to stream the application from the Configuration Manager cache into the App-V cache. If you deploy a virtual application to client computers and its content is not in the App-V cache, the App-V Client streams the application content from the Configuration Manager client cache into the App-V cache, and then runs the application. After the application runs successfully, you can set the Configuration Manager client to delete any older versions of the package at the next deletion cycle, or to persist them in Configuration Manager client cache. Persisting content locally can take advantage of package content delivery optimization methods such as BranchCache and PeerCache.

Use the information in this table to help you decide if local delivery is the best delivery method for you:

Advantages	Disadvantages
<p>The standard distribution point functionality is used to download the package by using Background Intelligent Transfer Service (BITS).</p> <p>Virtual application package contents are delivered locally to the client. This means that users can run them when their computer is not connected to the network.</p> <p>This method is suitable for slow or unreliable network connections and for computers that only occasionally connect to the network.</p> <p>Configuration Manager uses Remote Differential Compression (RDC) to send to clients only the bytes within the files that have changed when virtual application package content is updated. The Configuration Manager client uses RDC to build a new version of a virtual application package based on the current version of the package and any changes sent to the client.</p> <p>This method provides application resiliency for mobile users or disconnected users. Admins can choose to persist the package in the Configuration Manager cache after delivery if the virtual application was deployed with an install action. The package in the Configuration Manager client cache serves as a local, reliable streaming source for the App-V Client to pull the package into its cache.</p>	<p>Disk space that equals up to twice the size of the virtual application package is required on the client when the virtual application is persisted in the Configuration Manager cache.</p>

Deployment from an image

You can also preinstall virtual applications on a computer and then create an image of that computer for deployment to other computers. But if the virtual application package was created at a different site, the binary delta replication will not be used to download updates to the application. This option can be useful in a virtual desktop infrastructure when you want applications to be available immediately instead of downloading the applications after the user logs on.

Migrating from an App-V infrastructure to a Configuration Manager and App-V infrastructure

Use the following table to help you plan a migration from an existing App-V infrastructure to virtual application management with Configuration Manager.

Step	More information
Examine your current virtual applications to choose the applications that you want to migrate to your Configuration Manager infrastructure.	No additional information.
Evaluate the users and devices to which the virtual applications will be deployed.	Create Configuration Manager collections to group together the users and devices to which you want to deploy the virtual applications. See Introduction to collections .
Migrate App-V 5 connection groups to Configuration Manager virtual environments.	See the Migrate App-V 5 connection groups to Configuration Manager virtual environments section in this topic.
Investigate to find out if any of your virtual applications exist as full applications in your Configuration Manager infrastructure.	For easier management, you can add the virtual application as a new deployment type to the existing full application. See Create applications .
Create applications to replace your existing App-V packages.	See Introduction to application management and Create applications .
Configuration Manager begins to manage virtual applications on a client after the first deployment of a virtual application. After this, Configuration Manager must manage all App-V applications on the computer.	No additional information.

Step	More information
Distribute the content to the appropriate distribution points to enable local delivery of applications.	See Manage content and content infrastructure .
Deploy the application to Configuration Manager clients.	See Deploy applications .
<p>If the App-V application was created with an earlier version of the sequencer that does not create a manifest XML file, you can open it and save it in a newer version of the sequencer to create the file. This file is required to deploy virtual applications with Configuration Manager.</p>	
<p>App-V supports the virtual application packages that are created with the SoftGrid 4.1 SP1 or 4.2 versions of the sequencer.</p>	
<p>If the applications were previously installed locally, you must uninstall them before you deploy a virtual version of the application.</p>	
<p>Configuration Manager no longer supports using packages and programs that contain virtual applications. When you migrate from Configuration Manager 2007 to Configuration Manager current branch, Configuration Manager converts these packages into applications.</p>	<p>See Planning for the migration of objects to Configuration Manager current branch.</p>
<p>Configuration Manager 2007 advertisements are converted into the following deployment types:</p> <ul style="list-style-type: none"> - Migrating App-V packages with no advertisement: One deployment type that uses the default deployment type settings. 	
<ul style="list-style-type: none"> - Migrating App-V packages with one advertisement: One deployment type that uses the same settings as the Configuration Manager 2007 advertisement. 	
<ul style="list-style-type: none"> - Migrating App-V packages with multiple advertisements: A deployment type, for each Configuration Manager 2007 advertisement, that uses the settings for that advertisement. 	

Migrating App-V 5 connection groups to Configuration Manager virtual environments

App-V virtual environments in Configuration Manager allow virtual applications that you have deployed to share the same file system and registry on client computers. This means that unlike standard virtual applications, these applications can share data with each other. Virtual environments are created or changed on client computers when the application is installed or when clients next evaluate their installed applications. Virtual environments are similar to connection groups in standalone App-V 5.

When you migrate connection groups from standalone App-V 5 to Configuration Manager virtual environments, you must ensure that Configuration Manager correctly manages the connection groups that already exist on client computers, and that the user's environment within those connection groups is preserved.

To convert App-V 5 connection groups to Configuration Manager virtual environments:

1. Create Configuration Manager applications for all applications that existed in App-V.
2. Deploy the applications to users or devices with a deployment purpose of **Required**. Deployments to users must be deployed to the same users who used the application in App-V. Deployments to computers must be deployed to the same computers that had the application in App-V.
3. After the deployment is finished, create virtual environments that match the connection groups that are published in standalone App-V. The virtual environment must have the same packages (specifically, App-V 5 deployment types) in the same order.

For information about how to create an App-V virtual environment, see [How to create App-V virtual environments](#).

Alternatively, you can delete all connection groups from the App-V Client before you begin to deploy applications with Configuration Manager. But any settings that users might have saved in App-V connection groups will be lost.

Dynamic Suite Composition in App-V 4.6

Dynamic Suite Composition is a feature that lets you define one virtual application package as having a dependency on another virtual application package. When the application is run, the App-V Client hosts the primary package and the dependent package in the same virtual environment for the application.

For you to use this feature with Configuration Manager, both packages must be deployed and registered with the App-V Client. To ensure that dependent package

content is hosted locally on the client computer, set up the application deployment for local delivery (download and execute).

For more information about App-V Dynamic Suite Composition, see your App-V documentation.

Converting App-V 4.6 applications to App-V 5 applications

The application package format has changed between App-V 4.6 and App-V 5. Applications that have been sequenced by using App-V 4.6 are no longer supported. But App-V 5 has a package converter tool that you can use to convert applications. For more information, see [How to convert a package created in a previous version of App-V](#).

Use the following steps to convert App-V 4.6 applications to App-V 5 applications:

1. Convert or resequence the App-V 4.6 packages into the App-V 5 format.
2. Deploy the App-V 5 client to computers in your hierarchy.
3. Create new applications that contain deployment types for your App-V 5 applications, and create supersedence rules to supersede the App-V 4.6 applications.
4. Create virtual environments as required.
5. Deploy the new App-V 5 applications to computers.

User and deployment configuration files

User and deployment configuration files have settings that control how an application behaves. You can use these files to change application settings without resequencing the application.

A typical App-V 5 application might contain the following files:

- An application package (.appv) file
- A user configuration file
- A deployment configuration file

The user configuration file has settings that apply only to the logged-on user. You can, for example, edit the configuration files to change the information about the application

shortcut that will be deployed to users. You can also create a Configuration Manager application with multiple deployment types. Each deployment type can contain a different user configuration file and use requirement rules to ensure that these are installed for the relevant users.

The deployment configuration file has settings that apply to the computer, like registry settings. The file can also have user settings, which are applied to all users.

If you want to deploy App-V 5 virtual applications with Configuration Manager, all three files must be present in the same folder when you create the App-V 5 deployment type. If there are multiple files in the folder, Configuration Manager will use the most recent.

For more information, see your [About App-V 5.0 dynamic configuration](#).

App-V local interaction

In some application deployment scenarios, applications are installed locally on client computers, and other applications are deployed as virtual applications to the same client computer. By default, the applications that were locally installed cannot see or communicate directly with virtualized applications. This is the intended behavior of the application isolation that App-V provides. Local interaction is a feature of the App-V Client that you can enable for each application to allow locally installed applications that run on a client computer to see and communicate with virtualized applications.

Configuration Manager and App-V fully support local interaction.

For more information about the App-V local interaction feature, see your App-V documentation.

App-V 5 Shared Content Store

Configuration Manager supports the App-V 5 Shared Content Store feature. For more information, see [Planning for the App-V 5.0 Shared Content Store \(SCS\)](#).

Monitoring virtual applications

Virtual application reports

You can use the following reports to monitor App-V in your Configuration Manager environment:

Report name	Description
App-V Virtual Environment Results	Shows information about a selected virtual environment that is in a specified state for a selected collection (App-V 5 only).
App-V Virtual Environment Results For Asset	Shows information about a selected virtual environment for a specified asset and any deployment types for the selected virtual environment (App-V 5 only).
App-V Virtual Environment Status	Shows compliance information for a selected virtual environment for a selected collection. The Retained column in this report shows the assets in which a virtual environment that was previously set up is no longer applicable, but it is retained to persist user settings in applications that run in the virtual environment (App-V 5 only).
Computers with a specific virtual application	Shows a summary of computers that have the specified App-V shortcut that the Application Virtualization Management Sequencer created (App-V 4.6 only).
Computers with a specific virtual application package	Shows a list of computers that have the specified App-V application package installed (App-V 4.6 only).
Count all instances of virtual application packages	Shows a count of all detected App-V application packages (App-V 4.6 only).
Count all instances of virtual applications	Shows a count of all detected App-V applications (App-V 4.6 only).

Log files

Configuration Manager records information about virtual application deployments in log files. For information about the log files that virtual applications and Configuration

Manager application management use, see [Log files](#).

For Windows 8.1, find logs for the App-V client in
C:\ProgramData\Microsoft\Application Virtualization Client.

Disable and delete application deployments

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

If you want to stop the deployment of an application, you can either disable it temporarily or delete it entirely.

Important

Neither of these actions by themselves cause an instant change on the client. You can use [client notifications](#) or other automation tools to quickly request that clients refresh policy. But that still doesn't guarantee that a client won't run a deployment.

Make sure you carefully plan app deployments. [Simulate](#) more complex deployments. When you deploy to a query-based collection, use [query results preview](#) to make sure you understand the scope of the query.

Disable

Starting in version 2103, you can disable application deployments. Other objects already have similar behaviors:

- Software update deployments: Disable the deployment
- Phased deployments: Suspend the phase
- Package: Disable the program
- Task sequence: Disable the task sequence
- Configuration baseline: Disable the baseline

For device-based deployments, when you disable the deployment or object, use the client notification action to [Download Computer Policy](#). This action immediately tells the client to update its policy from the site. If the deployment hasn't already started, the client receives the updated policy that the object is now disabled.

For user-based deployments, the user needs to sign out of Windows. Policy updates when they sign in to Windows, or every 24 hours by default.

Note

You can't disable an available deployment of an application to a user collection. You can only disable required deployments to user collections, or both type of deployments to device collections. The following table summarizes the supported scenarios to disable app deployments:

Deployment purpose	Device collection	User collection
Required	Yes	Yes
Available	Yes	No

1. In the Configuration Manager console, go to the **Software Library** workspace, expand **Application Management**, and select the **Applications** node.
2. Select an app that you've deployed. In the details pane, switch to the **Deployment** tab.
3. Select a deployment. In the ribbon, on the **Deployment** tab, select **Disable**.
4. For a device-based deployment, note the name of the collection in **Collection** field of the deployment.

Tip

When you select the deployment, press **CTRL + C**. This keyboard shortcut copies the values of the current columns for the selected deployment.

5. Switch to the **Assets and Compliance** workspace, select the **Device Collections** node, and locate the target collection for the deployment. The quickest method is to search for the collection name as previously noted. You may need to select the option in the ribbon to search **All subfolders**.
6. Select the target collection for the deployment. In the ribbon, in the **Collection** group, select **Client Notification** and choose the **Download Computer Policy** action.

To enable the deployment, repeat this process but select the **Enable** action on the application deployment.

Note

When you select a deployment, you can use the **Collection** action to change to the **Assets and Compliance** workspace. But the current collection view doesn't support client notification actions.

Delete

1. In the Configuration Manager console, go to the **Software Library** workspace, expand **Application Management**, and select either the **Applications** or **Application Groups** node.
2. Select the application or application group that includes the deployment you want to delete.
3. Switch to the **Deployments** tab of the details pane, and select the deployment.
4. In the ribbon, on the **Deployment** tab in the **Deployment** group, select **Delete**.

When you delete an application deployment, any instances of the application that clients have already installed aren't removed. To remove these applications, deploy the application to computers to **Uninstall**. If you delete an application deployment, the application is no longer visible in Software Center. The same behavior happens when you remove a resource from the target collection for the deployment.

When you delete a deployment, you remove the policy that deploys an application to a specific collection. This action doesn't delete the collection, any deployment types, or the application itself.

Next steps

[Revise and supersede applications](#)

[Uninstall applications](#)

Troubleshooting tips for application deployments

Article • 10/28/2022

Applies to: Configuration Manager (current branch)

Typical problems with application deployments fall into one of the following categories:

- Application download failures
- Application deployment compliance stuck at 0%

If you experience either of these issues, this article provides some steps you can use to troubleshoot. For more in-depth troubleshooting, see [Troubleshooting application deployment technical reference](#).

Download failures

Application download failures include the following problems:

- The client is stuck downloading an application
- The client fails to download the application content
- The client gets stuck at 0% while downloading the application

The first thing to check when you experience application download failures is for missing or misconfigured boundaries and boundary groups. For example, if the client is on the intranet and not configured for internet-only client management, its network location must be in a configured boundary. There must also be a boundary group assigned to this boundary for the client to download content. For more information, see [Define site boundaries and boundary groups](#).

If you can't configure a boundary for a client, or if a specific boundary group can't be a member of another boundary group:

1. In the Configuration Manager console, open the properties of the **Deployment Type**.
2. Switch to the **Content** tab.
3. In the section for using a distribution point from a neighbor boundary group or the default site boundary group, change the **Deployment options** to **Download**

content from distribution point and run locally. (By default this setting is **Do not download content**.)

If the client can't download the application content, make sure it's distributed to a distribution point. To verify this configuration, use the in-console features to monitor content distribution to the distribution points. For more information, see [Monitor content you have distributed](#).

Compliance stuck at 0%

When the application's deployment compliance is 0%, check the deployment status for the application in the **Monitoring** workspace under the **Deployments** node.

- **In Progress:** The client could be stuck [downloading content](#)
- **Error:** For more information on how to troubleshoot this problem, see the following blog post: [Tips and Tricks: How to Take Action on Assets That Report a Failed Deployment ↗](#)
- **Unknown:** This status usually means that the client hasn't received policy. Manually refresh client policy to see if the client receives it. For more information, see [Initiate policy retrieval for a Configuration Manager client](#).

If these actions don't resolve the issue, check the client status. There may be a deeper underlying problem with the client. For more information, see [How to monitor clients](#).

Next steps

- [Monitor applications](#)
- [Deploy applications](#)
- [Management tasks for applications](#)
- [Troubleshooting application deployment technical reference](#)

Feedback

Was this page helpful?



[Provide product feedback ↗](#) | [Get help at Microsoft Q&A](#)

Monitor applications from the Configuration Manager console

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Applications in Configuration Manager support state-based monitoring, which enables you to track the last application deployment state for users and devices. These state messages display information about individual devices. For example, if an application is deployed to a collection of users, you can view the compliance state of the deployment and the deployment purpose in the Configuration Manager console.

Monitor application deployments by using the **Monitoring** workspace in the Configuration Manager console or by using reports.

About compliance states

An application deployment state has one of the following compliance states:

- **Success:** The client successfully deployed the application or it's already deployed. A deployment can install or uninstall an application.
- **In progress:** The client is currently running the application deployment.
- **Unknown:** Configuration Manager can't determine the state of the application deployment. This state isn't applicable for deployments with a purpose of **Available**. The console typically displays this state when the site hasn't yet received state messages from the client.
- **Requirements not met:** The client didn't run the application deployment because it wasn't compliant with a dependency or a requirement rule. For example, the OS on the device isn't applicable.
- **Error:** The client failed to deploy the application because of an error.

For each compliance state, you can view additional information, such as the number of users and devices in this category. The compliance states also include subcategories. For example, the **Error** compliance state includes the following subcategories:

- Error evaluating requirements
- Content related errors

- Installation errors

When more than one compliance state applies for an application deployment, you can see the aggregate state that represents the lowest compliance. For example:

- A user signs in to two devices. The application successfully installs on one device but fails to install on the other. The aggregate deployment state of the application for this user displays as **Error**.
- You deploy an application to all users that sign in to a computer. Configuration Manager displays multiple deployment results for that computer. If one of the deployments fails, the aggregate deployment state for the computer displays as **Error**.

Use these subcategories to help you to quickly identify any important issues with an application deployment. You can also view additional information about the devices that fall into a particular subcategory of a compliance state.

Application monitoring reports

Application management in Configuration Manager includes many built-in reports to monitor information about applications and deployments. These reports have the report category of **Software Distribution – Application Monitoring**. For more information, see [List of reports](#).

For more information about how to configure reporting in Configuration Manager, see [Introduction to reporting](#).

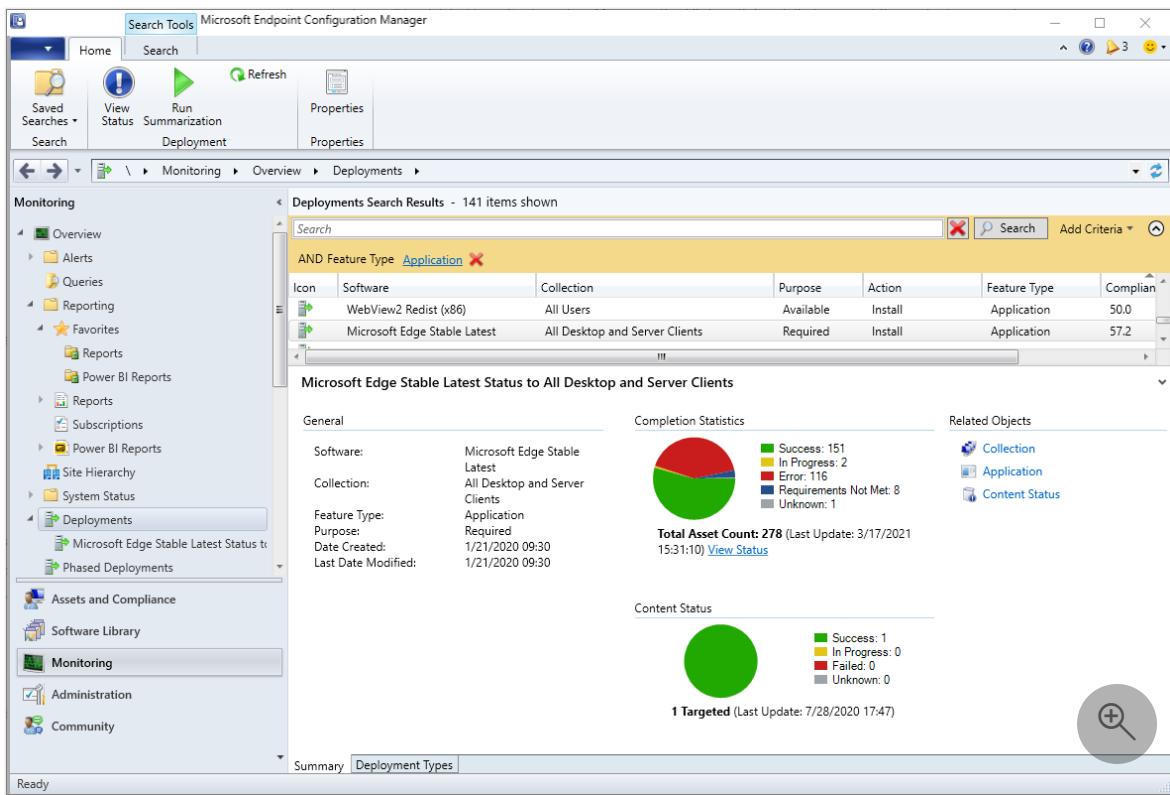
Monitor app state in the console

1. In the Configuration Manager console, go to the **Monitoring** workspace, and select the **Deployments** node.

If your site has numerous deployments, filter the list to just application deployments.

- a. At the top of the list next to the *Search* field, select **Add Criteria**. Choose **Feature Type** and then select **Add**.
- b. The search area adds the default criteria, **AND Feature Type Content Distribution**. Select **Content Distribution** and choose **Application** instead.
- c. Select **Search** to refresh the list.

2. Select a deployment for the app to monitor.



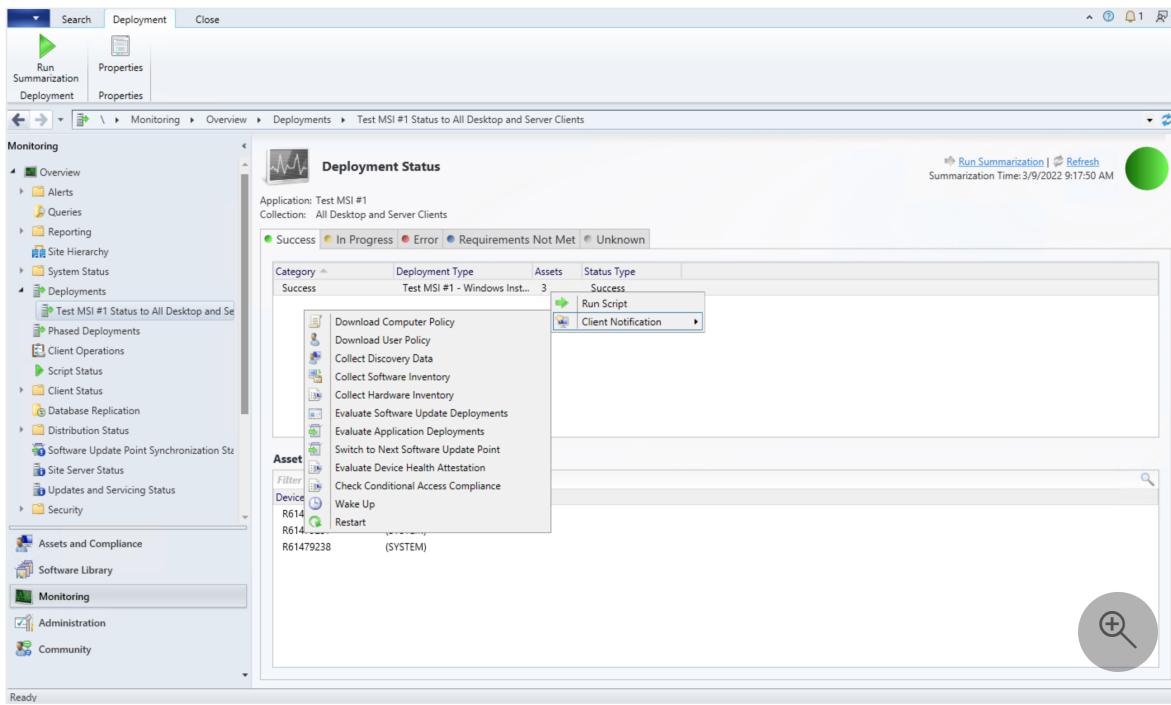
The following two tabs in the details pane are populated for the selected deployment:

- **Summary:** Displays general status information about an application deployment.
- **Deployment Types:** Displays status for the application's deployment types.

Review deployment details

From the **Deployments** node, you can review deployment details for each compliance state and the resources in that state. To review the deployment details, select **View status** on the **Home** tab of the ribbon. This action opens the **Deployment Status** pane. Here you can review the assets in each compliance state. To display Details list. Then select **More Details** on the right side of the window.

- The maximum number of items that the **Deployment Status** pane can display is 20,000. If you need to see more items, use Configuration Manager reports to review application status data.
- The status of deployment types is aggregated in the **Deployment Status** pane. To display more detailed information about the deployment types, use the **Application Infrastructure Errors** report.
- Starting in version 2203, you can perform client notification actions, including **Run Scripts**, from the **Deployment Status** view. Use the right-click menu on either a group of clients in a **Category** or a single client in the **Asset details** pane to display the client notification actions.



Summarized data

The information on the **Summary** and **Deployment Types** tabs is summarized data. When you select **View Status**, the console displays current data from the site database. If these data don't match, select **Run Summarization**.

To configure the default application deployment summarization interval:

1. In the Configuration Manager console, go to the **Administration** workspace, expand **Site Configuration**, and select the **Sites** node.
2. Select the site for which you want to configure the summarization interval. Then in the **Settings** group of the ribbon, choose **Status Summarizers**.
3. Select **Application Deployment Summarizer**, and then select **Edit**.
4. Configure the summarization intervals:
 - **Frequency of status updates for a deployment that was modified in the last 30 days:** By default, this value is **60 minutes**.
 - **Frequency of status updates for a deployment that was modified in the last 31 to 90 days:** By default, this value is **24 hours**.
 - **Frequency of status updates for a deployment that was last modified over 90 days ago:** By default this value is **7 days**.

! Note

These values apply to application, task sequence, and package deployments.

The site calculates the period of time based on the deployment start time.

Next steps

[Monitor phased deployments](#)

[Monitor app usage with software metering](#)

Manage and monitor phased deployments

Article • 10/04/2022

This article describes how to manage and monitor phased deployments. Management tasks include manually beginning the next phase, and suspend or resume a phase.

First, you need to create a phased deployment:

- [Application](#)
- [Software update](#)
- [Task sequence](#)

Move to the next phase

When you select the setting, **Manually begin the second phase of deployment**, the site doesn't automatically start the next phase based on success criteria. You need to move the phased deployment to the next phase.

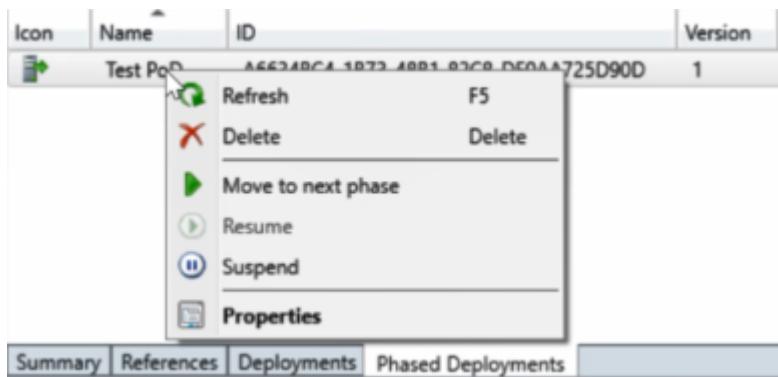
1. How to start this action varies based on the type of deployed software:

- **Application:** Go to the **Software Library** workspace, expand **Application Management**, and select **Applications**.
- **Software update:** Go to the **Software Library** workspace, and then select one of the following nodes:
 - **Software Updates**
 - **All Software Updates**
 - **Software Update Groups**
 - **Windows Servicing, All Windows Updates**
 - **Office 365 Client Management, Office 365 Updates**
- **Task sequence:** Go to the **Software Library** workspace, expand **Operating Systems**, and select **Task Sequences**.

2. Select the software with the phased deployment.

3. In the details pane, switch to the **Phased Deployments** tab.

4. Select the phased deployment, and click **Move to next phase** in the ribbon.



Optionally, use the following Windows PowerShell cmdlet for this task: [Move-CMPhasedDeploymentToNext](#).

Suspend and resume phases

You can manually suspend or resume a phased deployment. For example, you create a phased deployment for a task sequence. While monitoring the phase to your pilot group, you notice a large number of failures. You suspend the phased deployment to stop further devices from running the task sequence. After resolving the issue, you resume the phased deployment to continue the rollout.

1. How to start this action varies based on the type of deployed software:
 - **Application:** Go to the **Software Library** workspace, expand **Application Management**, and select **Applications**.
 - **Software update:** Go to the **Software Library** workspace, and then select one of the following nodes:
 - Software Updates
 - **All Software Updates**
 - **Software Update Groups**
 - Windows Servicing, **All Windows Updates**
 - Office 365 Client Management, **Office 365 Updates**
 - **Task sequence:** Go to the **Software Library** workspace, expand **Operating Systems**, and select **Task Sequences**. Select an existing task sequence, and then click **Create Phased Deployment** in the ribbon.
2. Select the software with the phased deployment.
3. In the details pane, switch to the **Phased Deployments** tab.
4. Select the phased deployment, and click **Suspend** or **Resume** in the ribbon.

Note

Starting on April 21, 2020, Office 365 ProPlus is being renamed to **Microsoft 365 Apps for enterprise**. For more information, see [Name change for Office 365 ProPlus](#). You may still see the old name in the Configuration Manager product and documentation while the console is being updated.

Optionally, use the following Windows PowerShell cmdlets for this task:

- [Suspend-CMPhasedDeployment](#)
- [Resume-CMPhasedDeployment](#)

Monitor

Phased deployments have their own dedicated monitoring node, making it easier to identify phased deployments you have created and navigate to the phased deployment monitoring view. From the **Monitoring** workspace, select **Phased Deployments**, then double-click one of the phased deployments to see the status.

Phased Deployment Status - MyTest Phased Deployment 4 Status

Phased Deployment Status - MyTest Phased Deployment 4

Software 2019-04 Cumulative Update for Windows Server 2016 for x64-based Systems (KB4493470)

Type Software Update

Select Phase: **Phase 1** ▾

Phase Name	Total Resources	Status	Progress
Phase 1	1	Deployment Created	<div style="width: 100%; background-color: green;"></div>
Phase 2	3	Waiting	

Success Criteria

Deployment Success
Phase Goal 65%



100.0%

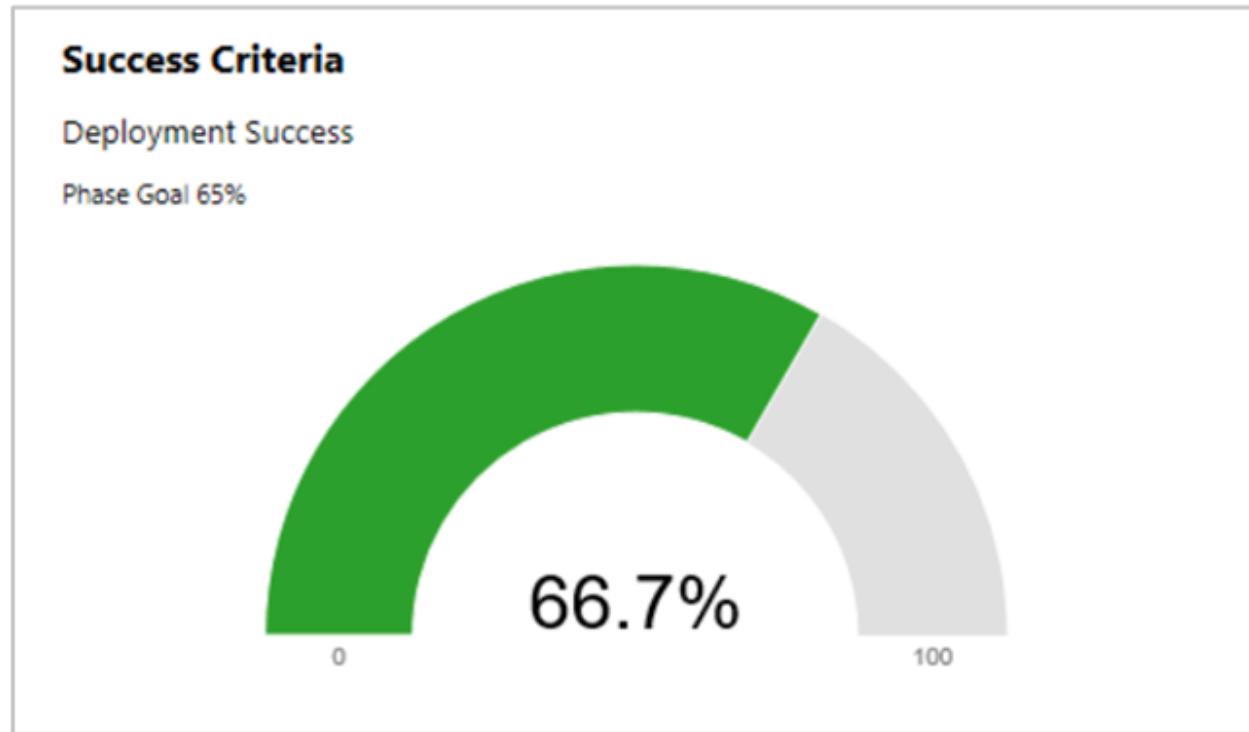
This dashboard shows the following information for each phase in the deployment:

- **Total devices or Total resources:** How many devices are targeted by this phase.
- **Status:** The current status of this phase. Each phase can be in one of the following states:
 - **Deployment created:** The phased deployment created a deployment of the software to the collection for this phase. Clients are actively targeted with this software.
 - **Waiting:** The previous phase hasn't yet reached the success criteria for the deployment to continue to this phase.
 - **Suspended:** An administrator suspended the deployment.
- **Progress:** The color-coded deployment states from clients. For example: Success, In Progress, Error, Requirements Not Met, and Unknown.

Success criteria tile

Use the **Select Phase** drop-down list to change the display of the **Success Criteria** tile. This tile compares the **Phase Goal** against the current compliance of the deployment. With the default settings, the phase goal is 95%. This value means that the deployment needs a 95% compliance to move to the next phase.

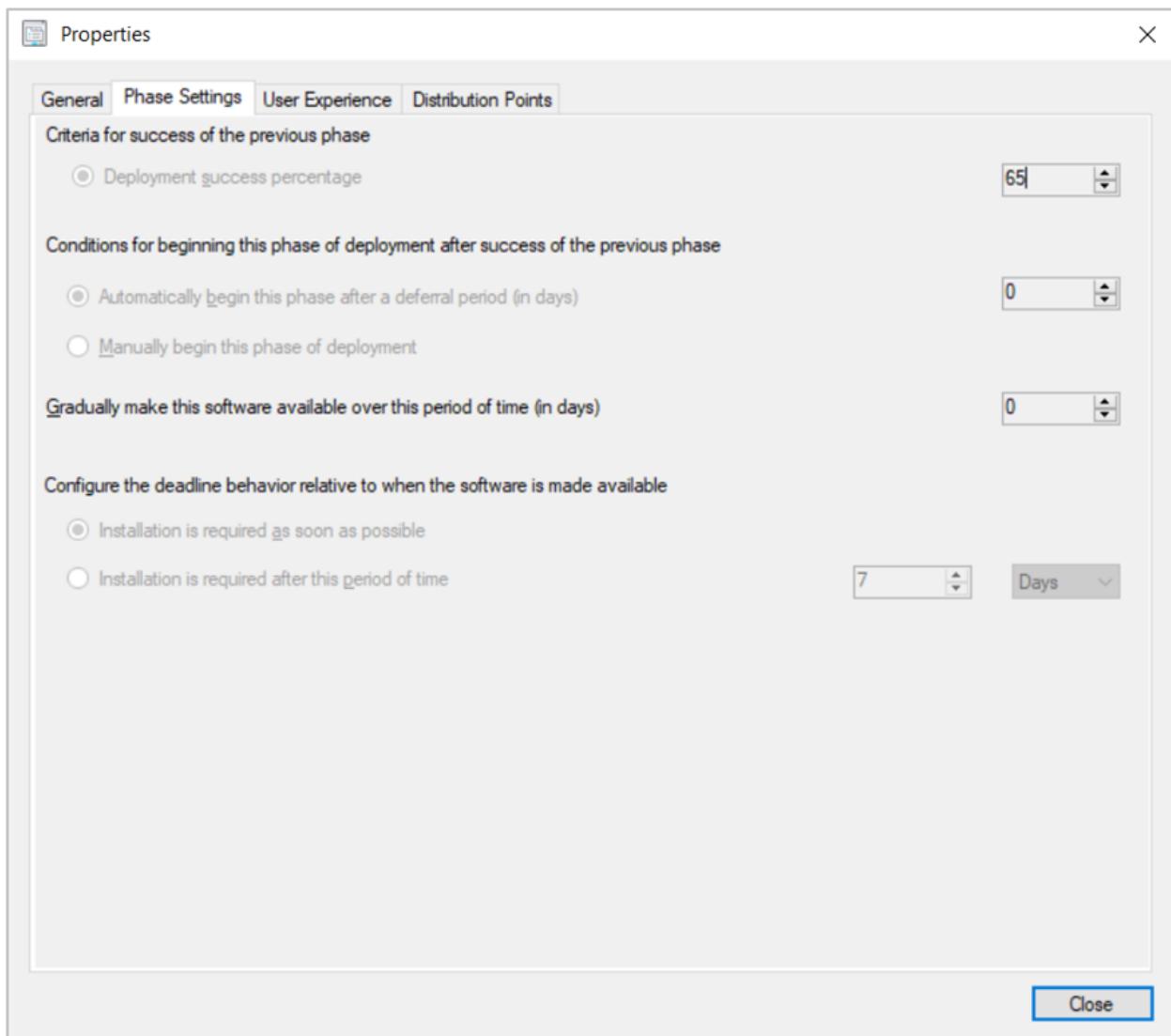
In the example, the phase goal is 65%, and the current compliance is 66.7%. The phased deployment automatically moved to the second phase, because the first phase met the success criteria.



The phase goal is the same as the **Deployment success percentage** on the Phase Settings for the *next* phase. For the phased deployment to start the next phase, that second phase defines the criteria for success of the first phase. To view this setting:

1. Go to the phased deployment object on the software, and open the Phased Deployment Properties.
2. Switch to the **Phases** tab. Select **Phase 2** and click **View**.
3. In the phase Properties window, switch to the **Phase Settings** tab.
4. View the value for **Deployment success percentage** in the *Criteria for success of the previous phase* group.

For example, the following properties are for the same phase as the success criteria tile shown above where the criteria is 65%:



PowerShell

Use the following Windows PowerShell cmdlets to manage phased deployments:

Automatically create phased deployments

- [New-CMApplicationAutoPhasedDeployment](#)
- [New-CMSoftwareUpdateAutoPhasedDeployment](#)
- [New-CMTaskSequenceAutoPhasedDeployment](#)

Manually create phased deployments

- [New-CMSoftwareUpdatePhase](#)
- [New-CMSoftwareUpdateManualPhasedDeployment](#)
- [New-CMTaskSequencePhase](#)
- [New-CMTaskSequenceManualPhasedDeployment](#)

Get existing phased deployment objects

- [Get-CMApplicationPhasedDeployment](#)
- [Get-CMSoftwareUpdatePhasedDeployment](#)
- [Get-CMTaskSequencePhasedDeployment](#)
- [Get-CMPhase](#)

Monitor phased deployment status

- [Get-CMPhasedDeploymentStatus](#)

Manage existing phased deployments

- [Move-CMPhasedDeploymentToNext](#)
- [Resume-CMPhasedDeployment](#)
- [Suspend-CMPhasedDeployment](#)

Modify existing phased deployments

- [Set-CMApplicationPhasedDeployment](#)
- [Set-CMSoftwareUpdatePhase](#)
- [Set-CMSoftwareUpdatePhasedDeployment](#)
- [Set-CMTaskSequencePhase](#)
- [Set-CMTaskSequencePhasedDeployment](#)
- [Remove-CMApplicationPhasedDeployment](#)
- [Remove-CMSoftwareUpdatePhasedDeployment](#)
- [Remove-CMTaskSequencePhasedDeployment](#)

Software metering in Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

This topic contains a reference for all of the operations you might perform when using Configuration Manager software metering.

i Important

Software metering is used to monitor Windows PC desktop apps with a filename ending in .exe. Software metering does not monitor modern Windows apps (such as those used by Windows 8).

Prerequisites for software metering

Software metering has no external dependencies, only dependencies within the product.

Dependency	More information
Client settings for software metering.	To use software metering, the client setting Enable software metering on clients must be enabled and deployed to computers. You can deploy software metering settings to all computers in the hierarchy, or you can deploy custom settings to groups of computers. See Configure software metering in this topic.
The reporting services point.	You must configure a reporting services point before you can view software metering reports. For more information, see Introduction to reporting .

Configure software metering

This procedure configures the default client settings for software metering and applies to all computers in your hierarchy. If you want these settings to apply to only some computers, create a custom device client setting and deploy it to a collection that contains the computers on which you want to use software metering. For more information about how to create custom device settings, see [Configure client settings](#).

1. In the Configuration Manager console, click **Administration > Client Settings > Default Client Settings**.
2. On the **Home** tab, in the **Properties** group, click **Properties**.
3. In the **Default Settings** dialog box, click **Software Metering**.
4. In the **Device Settings** list, configure the following:
 - **Enable software metering on clients:** Select **True** to enable software metering.
 - **Schedule data collection:** Configure how often software metering data is collected from client computers. Use the default value of every **7 days** or click **Schedule** to specify a custom schedule.
5. Click **OK** to close the **Default Settings** dialog box.

Client computers are configured with these settings the next time they download client policy. To initiate policy retrieval for a single client, see [Manage clients](#).

Create software metering rules

Use the Create Software Metering Rule wizard to create a new software metering rule for your Configuration Manager site.

1. In the Configuration Manager console, click **Assets and Compliance > Software Metering**.
2. On the **Home** tab, in the **Create** group, click **Create Software Metering Rule**.
3. On the **General** page of the Create Software Metering Rule wizard, specify the following information:
 - **Name** - The name of the software metering rule. This should be unique and descriptive.

Note

Software metering rules can share the same name if the file name contained in the rules is different.

- **File Name** - The name of the program file that you want to meter. You can click **Browse** to display the **Open** dialog box, in which you can select the

program file to use.

 **Note**

If you type the executable file name in the **File name** box, no checks are carried out to determine whether this file exists or whether it contains the necessary header information. When possible, click **Browse** and select the executable file to be metered.

Wildcard characters are not permitted in the file name.

This box is optional if a value for **Original file name** is specified.

- **Original File Name** - The name of the executable file that you want to meter. This name matches information in the header of the file, not the file name itself so that it can be useful in cases where the executable file has been renamed but you want to meter it by the original name.

 **Note**

Wildcard characters are not permitted in the original file name.

This box is optional if a value for **File Name** is specified.

- **Version** - The version of the executable file you want to meter. You can use the wildcard character (*) to represent any string of characters or the wildcard character (?) to represent any single character. If you want to meter for all versions of an executable file, use the default value (*).
- **Language** - The language of the executable file to meter. The default value is the current locale of the operating system you are using. If you select an executable file to be metered by clicking the **Browse** button, this box is automatically filled if language information is present in the header of the file. To meter all language versions of a file, select **Any** in the drop-down list.
- **Description** - An optional description for the software metering rule.
- **Apply this software metering rule to the following clients** – Select whether you want to apply the software metering rule to all clients in the hierarchy or to the clients that are assigned to the site specified in the **Site** list.

4. To continue, click **Next**.

5. Review and confirm the settings and then complete the wizard to create the software metering rule. The new software metering rule is displayed in the **Software Metering** node in the **Assets and Compliance** workspace.

Configure automatic software metering rules

You can configure software metering in Configuration Manager to automatically generate disabled software metering rules from recent usage inventory data held in the site database. You can configure this inventory data so that only for applications that are used on a specified percentage of computers metering rules are created. You can also specify the maximum number of automatically generated software metering rules allowed on the site.

 **Note**

By default, software metering rules that are automatically created are disabled. Before you can begin to collect usage data from these rules, you must enable them.

1. In the Configuration Manager console, click **Assets and Compliance > Software Metering**, and then, in the **Home** tab, in the **Settings** group, click **Software Metering Properties**.
2. In the **Software Metering Properties** dialog box, configure the following:
 - **Data retention (in days)** - Specifies the amount of time that data generated by software metering rules are kept in the site database. The default value is **90** days.
 - Enable the option **Automatically create disabled metering rules from recent usage inventory data**.
 - **Specify the percentage of computers in the hierarchy that must use a program before a software metering rule is automatically created** - The default value is **10** percent.
 - **Specify the number of software metering rules that must be exceeded in the hierarchy before the automatic creation of rules is disabled** - The default value is **100** rules.
3. Click **OK** to close the **Software Metering Properties** dialog box.

Manage software metering rules

In the **Assets and Compliance** workspace, select **Software Metering**, select the software metering rule to manage, and then select a management task.

Use the following table for more information about the management tasks that might require some information before you select them.

Management Details	
Task	
Enable	Enables or disables a software metering rule. This setting is downloaded to client computers according to the Client policy polling interval in the Client Policy section of client settings (by default, every 60 minutes).
Disable	See Configure client settings .

Monitor software metering

Software metering in Configuration Manager includes a number of built-in reports which allow you to monitor information about software metering operations. These reports have the report category of **Software Metering**.

For more information about how to configure reporting in Configuration Manager, see [Introduction to reporting](#).

Additionally, you can create queries and collections based on the data stored in the Configuration Manager database by software metering.

For more information about collections in Configuration Manager, see [Introduction to collections](#).

For more information about queries in Configuration Manager, see [Introduction to queries](#).

Security and privacy for software metering

Security Issues for Software Metering

An attacker could send invalid software metering information to Configuration Manager, which will be accepted by the management point even when the software metering client setting is disabled. This might result in a large number of metering rules that are

replicated throughout the hierarchy, causing a denial of service on the network and to Configuration Manager site servers.

Because an attacker can create invalid software metering data, do not consider software metering information to be authoritative.

Software metering is enabled by default as a client setting.

Privacy Information for Software Metering

Software metering monitors the usage of applications on client computers. Software metering is enabled by default. You must configure which applications to meter.

Metering information is stored in the Configuration Manager database. The information is encrypted during transfer to a management point but it is not stored in encrypted form in the Configuration Manager database.

This information is retained in the database until it is deleted by the site maintenance tasks **Delete Aged Software Metering Data** (every five days) and **Delete Aged Software Metering Summary Data** (every 270 days). You can configure the deletion interval. Metering information is not sent to Microsoft.

Before you configure software metering, consider your privacy requirements.

Example scenario for using software metering

In this section, you'll create an example software metering rule that can help you solve the following business requirements:

- Determine how many copies of a particular app are in your company
- Discover any unused copies of an app
- Determine which users regularly use a particular app

Woodgrove Bank has deployed Microsoft Office 2010 as its standard office productivity suite. However, to support a legacy application, some computers must continue to run Microsoft Office Word 2003. The IT department wants to reduce support and licensing costs by removing these copies of Word 2003 if the legacy application is no longer used. The help desk also wants to identify which users use the legacy application.

Woodgrove Bank's IT Systems Manager uses software metering in Configuration Manager to achieve these business objectives. The Admin performs the following

actions:

- Checks the prerequisites for software metering and confirms that the reporting services point is installed and operational.
- Configures the default client settings for software metering:
The Admin enables software metering and uses the default data collection schedule of once every seven days.
The Admin configures software inventory to inventory files that have the extension .exe by configuring the software inventory client setting **Inventory these file types**.
The Admin adds a new software metering rule, named **woodgrove.exe**, to monitor the legacy application.
- Waits for seven days, after which the client computers begin to report usage data for the **woodgrove.exe** executable.
- The Admin uses the Configuration Manager report **Install base for all metered software programs** to see which computers have the application **woodgrove.exe** loaded.
- After six months, the Admin runs the report **Computers that have a metered program installed, but have not run the program since a specified date**, specifying the software metering rule and a date six months in the past. This report identifies 120 computers that have not run the program in the past six months.
- The Admin makes some further checks to confirm that the legacy application is not required on the identified computers. The Admin then uninstalls the legacy application and the copy of Word 2003 from these computers.
The Admin runs the report **Users that have run a specific metered software program** to provide the help desk with a list of users who continue to use the legacy application.
- The Admin continues to check the software metering reports weekly and takes remedial action if necessary.

As a result of this course of action, IT support and licensing costs are reduced by removing the applications that are no longer required. In addition, the help desk now has the list that it wanted of the users who run the legacy application.

Management tasks for Configuration Manager applications

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Use the information in this article to help you manage Configuration Manager applications and deployment types.

For more information on how to create applications and deployment types, see [Create applications](#).

ⓘ Important

Depending on the type of application or deployment type, some management options might not be available.

Manage applications

In the **Software Library** workspace, expand **Application Management**, and select the **Applications** node. Select the application to manage, and then choose a management task in the ribbon.

Manage access accounts

Use this action to control access to the associated content on distribution points.

When you **Add** an account:

1. Specify one of the following account types:

- **User:** Any account that Windows can authenticate.
- **Guest:** An unauthenticated user.
- **Administrator:** An account that Windows recognizes as an administrator.
- **Windows User:** A specific user account. It can either be from a local machine or Active Directory.

2. Specify one of the following access rights:

- **No access:** Explicitly block the specified account type from accessing the content associated with this application.
- **Read**
- **Change**
- **Full control**

By default, the **Administrator** type has **Full control** access, and the **User** type has **Read** access.

Create prestaged content file

Prestaged content files help you to manage the delivery of content to remote distribution points. When scheduling and throttling options don't provide a valid solution for the remote distribution point, you can prestage the content.

For more information, see [Deploy and manage content](#).

Revision history

View and manage the revisions to this application. For more information, see [How to revise and supersede applications](#).

Update statistics

Updates the information that's displayed in the **Deployments** node of the **Monitoring** workspace about the deployments of this application. For more information, see [Monitor applications from the Configuration Manager console](#).

Create deployment type

Add a new deployment type to the selected application. For more information, see [Create deployment types for the application](#).

Convert to .MSIX

Convert an existing Windows Installer (.msi) application to the MSIX format. For more information, see [Support for MSIX format](#).

Reinstate

If you previously [retired](#) an application, use this action to reinstate it. When you reinstate a retired app, you can then deploy it again.

Retire

When you retire an application, it's no longer available for deployment. Configuration Manager doesn't delete the application and any deployments. If the app was installed on clients, Configuration Manager doesn't remove the app. Configuration Manager deletes any revisions to the app after 60 days in retirement.

Before you delete an application:

1. Retire the application.
2. Delete all deployments.
3. Remove references to the application by other deployments
4. Delete all of the application's revisions.

For more information, see [Revise and supersede applications](#).

Export

Export the selected applications to a .zip file that you can archive or import to another site. If you choose to export application content, Configuration Manager creates a folder with the content.

You can export:

- Application dependencies
- Supersedence relationships and conditions
- Content for the application and its dependencies

To automate this process, use the following Configuration Manager PowerShell cmdlets:

- [Export-CMApplication](#)
- [Import-CMApplication](#)

For more information, see [Import and export applications](#).

Copy (application)

Duplicate the application to create a new one. This action is useful to test something or when you need to create a similar application. The site creates a new application, and

appends `-copy` to the name. While the site copies most of the metadata to the new application, it doesn't copy any deployments.

Delete (application)

Delete the currently selected applications.

You can't delete an application if any of the following conditions are true:

- Other applications are dependent on it
- It has an active deployment
- It has dependent task sequences

Before you delete an application, [retire it](#).

Simulate deployment

Test the results of an application deployment to computers without installing or uninstalling it. For more information, see [Simulate application deployments](#).

Deploy

Deploy the selected application to a collection of computers. For more information, see [Deploy applications](#).

Create phased deployment

Phased deployments automate a coordinated, sequenced rollout of software across multiple collections. For example, deploy software to a pilot collection, and then automatically continue the rollout based on success criteria. For more information, see [Create phased deployments](#).

Distribute content

Copy the content for the selected application to distribution points. For more information, see [Distribute content](#).

Move

Move the selected application to another folder in the **Applications** node.

Set security scopes

Select the security scopes for the selected application. For more information, see [Security scopes](#).

Categorize

Administrative categories help you organize apps in the Configuration Manager console. You can add the **Administrative categories** column to the **Applications** node.

With this action, you can:

- Quickly add the selected app to an administrative category.
- Clear all categories on the current app.
- Select **Manage categories** to create, rename, or delete categories.

You can also manage categories on the application properties, **General information** tab.

Tip

To help users find apps by category in Software Center, define **user categories** for your apps. You can add these categories on the application properties, **Software Center** tab.

View relationships

Show a graphical diagram of the relationships of the selected applications to other applications. Choose one of the following relationship types:

- **Dependency:** Shows applications that are dependent on the selected application and the applications that the selected application depends on. For more information, see [Deployment type Dependencies](#).
- **Supersedence:** Shows applications that the selected application supersedes, and applications that the selected application is superseded by. For more information, see [Supersedence](#).
- **Global Conditions:** Shows the global conditions that this application references. For more information, see [Create global conditions](#).

Properties

Display and edit the metadata for this application.

Manage deployment types

In the **Software Library** workspace, expand **Application Management**, and select the **Applications** node. Select the application with the deployment type that you want to manage. In the details pane, switch to the **Deployment Types** tab. Select the deployment type that you want to manage, and then choose a management task from the **Deployment Type** tab of the ribbon.

Increase priority

Increase the priority of the selected deployment type. The Configuration Manager client evaluates deployment types in order. If the device meets the deployment type's requirements, it runs the deployment type. Then the client doesn't evaluate any further deployment types on the priority list.

Decrease priority

Lower the priority of the selected deployment type.

Copy (deployment type)

Duplicate the deployment type to create a new one. This action is useful to test something or when you need to create a similar deployment type. The site creates a new deployment type on the same application, and appends **-copy** to the name.

Delete (deployment type)

Delete the selected deployment type. You can't delete a deployment type if it's referenced by a deployment type in another application.

To delete a deployment type:

1. Remove all dependencies from other deployment types.
2. Remove previous revisions of all applications that have a deployment type that references this deployment type.

Update content

Refresh the content for the selected deployment type. When you refresh the content of a deployment type, the site creates a new revision of the application. This behavior might cause client devices to update with the new application content.

Next steps

[Import and export applications](#)

[Revise and supersede applications](#)

[Uninstall applications](#)

Link users and devices with user device affinity in Configuration Manager

Article • 02/22/2023

Applies to: Configuration Manager (current branch)

User device affinity in Configuration Manager associates a user with one or more devices. This behavior can eliminate the need to know the names of a user's devices to deploy an application to the user. Instead of deploying the application to each of the user's devices, you deploy the application to the user. Then, user device affinity automatically makes sure that the application installs on all devices that are associated with that user.

Define primary devices that users use every day for their work. When you create an affinity between a user and a device, you gain more app deployment options. For example, if a user requires Microsoft Visio, you can install it on the user's primary device by using a Windows Installer deployment. However, on a device that's not a primary device, you might deploy Visio as a virtual application. You also can use user device affinity to predeploy software on a user's device when the user isn't signed in. Then when the user logs on, the app is already installed and ready to run.

You only manage user device affinity information for computers. Configuration Manager automatically manages user device affinities for the mobile devices that it enrolls.

Manually set up user device affinity

1. In the Configuration Manager console, go to the **Assets and Compliance** workspace, and select the **Devices** node.
2. Select a device. On the **Home** tab in the ribbon, in the **Device** group, choose **Edit Primary Users**.
3. In the **Edit Primary Users** dialog box, search for and then select the users to add as primary users for the selected device. Choose **Add**.

Note

The **Primary Users** list shows users who are already primary users of this device, and the method by which each user-device relationship was assigned.

Set up primary devices for a user

1. In the Configuration Manager console, go to the **Assets and Compliance** workspace, and select the **Users** node.
2. Select a user. On the **Device** tab in the ribbon, choose **Edit Primary Devices**.
3. In the **Edit Primary Devices** dialog box, search for and then select the devices to add as primary devices for the selected user. Choose **Add**.

 **Note**

The **Primary Devices** list shows devices that are already set up as primary devices for this user, and the method by which each user-device relationship was assigned.

Automatically create user device affinities (Windows PCs only)

Configuration Manager reads data about user logon events from the Windows event log. To automatically create user device affinities, turn on these two options in the local security policy on client computers to store logon events in the Windows event log:

- Audit account logon events
- Audit logon events

To configure these settings, use Windows Group Policy.

 **Important**

If an error causes the Windows event log to generate a high number of entries, it might create a new event log. If this behavior occurs, existing logon events might not be available to Configuration Manager.

Set up the site to automatically create user device affinities

1. In the Configuration Manager console, go to the **Administration** workspace, and select the **Client Settings** node.

2. To modify the default client settings, select **Default Client Settings**. On the **Home** tab in the ribbon, in the **Properties** group, choose **Properties**. If you modify the default client settings, the site deploys them to all computers in the hierarchy. For more information, see [How to configure client settings](#).

- To create custom client agent settings, on the **Home** tab in the ribbon, in the **Create** group, choose **Create Custom Client Device Settings**.

3. In the **User and Device Affinity** group, set the following settings:

- **User device affinity threshold (minutes)**: Set the number of minutes of device usage before the site creates a user device affinity.
- **User device affinity threshold (days)**: Set the number of days over which the site measures the usage-based affinity threshold.
- **Automatically configure user device affinity from usage data**: Select **True** to let the site automatically create user device affinities. If you select **False**, you need to manually approve all user device affinity assignments.

As an example, if you set **User device affinity threshold (minutes)** to 60 minutes and you set **User device affinity threshold (days)** to 5 days, the user must use the device for at least 60 minutes over a period of 5 days to automatically create a user device affinity.

After Configuration Manager creates an automatic user device affinity, it continues to monitor the user device affinity thresholds. If the user's activity for the device falls below the thresholds you've set, the site removes the user device affinity. Set **User device affinity threshold (days)** to a value of at least seven days. This configuration avoids situations in which an automatically configured user device affinity might be lost while the user isn't signed in, for example, during the weekend.

 **Note**

Starting in Configuration Manager version 2010, the troubleshooting portal in the [Microsoft Intune admin center](#) allows you to search for a user and view their associated devices. Tenant attached devices that are assigned user device affinity automatically based on usage are returned when searching for a user. For more information, see [Tenant attach: ConfigMgr client details in the admin center](#).

Import user device affinities from a file

To create many relationships at one time, import a file that has the details for multiple user device affinities. Make sure the target devices are already discovered by the site and exist as resources in the Configuration Manager database.

1. In the Configuration Manager console, go to the **Assets and Compliance** workspace, and select either the **Users** or **Devices** node.
2. On the **Home** tab in the ribbon, in the **Create** group, choose **Import User Device Affinity**.
3. In the Import User Device Affinity Wizard, on the **Choose Mapping** page, set this information:
 - **File name.** Specify a comma-separated values (CSV) file that has a list of users and devices between which you want to create an affinity. In this file, each user-and-device pair must be on its own row, with values separated by a comma. Use this format: <domain>\<username>,<device NetBIOS name>
 - **This file has column headings for reference purposes.** If the .csv file has a top-row header, select this option. The site ignores the header row during the import.
4. If the file you import has more than two items in each row, use **Column** and **Assign** to specify which columns represent users and devices, and which columns to ignore during import.
5. Complete the wizard.

Let users create their own device affinities

Set up a user to create their own user device affinity in Software Center.

Set up the site to allow user-created user device affinity requests

- 1 In the Configuration Manager console, go to the **Administration** workspace, and select the **Client Settings** node.
 1. To modify the default client settings, select **Default Client Settings**. On the **Home** tab in the ribbon, in the **Properties** group, choose **Properties**.

To create custom client agent settings, on the **Home** tab in the ribbon, in the **Create** group, choose **Create Custom Client User Settings**.

Note

If you modify the default client settings, the site deploys them to all computers in the hierarchy. For more information, see [Configure client settings](#).

2. In the **User and Device Affinity** group, enable the setting to **Allow user to define their primary devices**.

Set up a user device affinity in Software Center

Users can use Software Center to set affinity.

1. In Software Center, go to the **Options** tab.
2. In the **Work information** section, select the option **I regularly use this computer to do my work**.

Manage user device affinity requests from users

When you disable the client setting to **Automatically configure user device affinity from usage data**, you need to manually approve all user device affinity assignments.

Approve or reject a user device affinity request

1. In the Configuration Manager console, go to the **Assets and Compliance** workspace.
2. Select the user or device collection for which you want to manage affinity requests.
3. On the **Home** tab in the ribbon, in the **Collection** group, choose **Manage Affinity Requests**.
4. In the **Manage User Device Affinity Requests** dialog box, select an affinity request, and then choose **Approve** or **Reject**.

Next steps

You can also use Microsoft Intune to find the primary use of an enrolled device. For more information, see [Find the primary user of an Intune device](#) in the Intune

documentation.

Manage apps from the Microsoft Store for Business and Education with Configuration Manager

Article • 04/11/2023

ⓘ Important

Starting in November 2021, this feature of Configuration Manager is **deprecated**. For more information, see [Update to Intune integration with the Microsoft Store on Windows ↗](#).

The [Microsoft Store for Business and Education](#) is where you find and acquire Windows apps for your organization. When you connect the store to Configuration Manager, you then synchronize the list of apps you've acquired. View these apps in the Configuration Manager console, and deploy them like you deploy any other app.

Online and offline apps

The Microsoft Store for Business and Education supports two types of app:

- **Online:** This license type requires users and devices to connect to the store to get an app and its license. Devices running Windows 10 or later should be Azure Active Directory (Azure AD)-joined or hybrid Azure AD-joined. They can also be [Azure AD-registered](#).
- **Offline:** This type lets you cache apps and licenses to deploy directly within your on-premises network. Devices don't need to connect to the store or have a connection to the internet.

For more information, see the [Microsoft Store for Business and Education overview](#).

Summary of capabilities

Configuration Manager supports managing Microsoft Store for Business and Education apps on devices running Windows 10 or later with the Configuration Manager client. Configuration Manager offers the following capabilities for online and offline apps:

Capability	Offline apps	Online apps
------------	--------------	-------------

Capability	Offline apps	Online apps
Synchronize app data to Configuration Manager (synchronization occurs every 24 hours)	Yes	Yes
Create Configuration Manager applications from store apps	Yes	Yes
Support for free apps from the store	Yes	Yes
Support for paid apps from the store	No	Yes <small>Note 1</small>
Support required deployments to user or device collections	Yes	Yes
Support available deployments to user or device collections	Yes	Yes
Support line-of-business apps from the store	Yes	Yes
Provision a store app for all users on a device <small>Note 2</small>	Yes	Yes

Note 1: Online licensed apps version requirement

To deploy online licensed apps to Windows devices with the Configuration Manager client, they need to be running a supported version of Windows 10 or later.

Note 2: Provision Windows app packages for all users on a device

For more information, see [Create Windows applications](#).

Deploying online apps using the Microsoft Store for Business and Education to devices that run the Configuration Manager client

Before deploying Microsoft Store for Business and Education apps to devices that run the full Configuration Manager client, consider the following points:

- For full functionality, devices need to be running a supported version of Windows 10 or later.
- Register or join devices to the same Azure AD tenant where you registered the Microsoft Store for Business and Education as a management tool.
- When the local Administrator account signs in on the device, it can't access Microsoft Store for Business and Education apps.

- Devices need a live internet connection to the Microsoft Store for Business and Education. For more information including proxy configuration, see [Prerequisites](#).

Set up synchronization

When you synchronize the list of Microsoft Store for Business and Education apps that your organization acquired, you see these apps in the Configuration Manager console.

Connect your Configuration Manager site to Azure AD and the Microsoft Store for Business and Education. For more information and details of this process, see [Configure Azure services](#). Create a connection to the **Microsoft Store for Business** service.

Make sure the service connection point and targeted devices can access the cloud service. For more information, see [Prerequisites for Microsoft Store for Business and Education - Proxy configuration](#).

Supplemental information and configuration

On the **App** page of the Azure Services Wizard, first configure the **Azure environment** and **Web app**. Then read the **More Information** section at the bottom of the page. This information includes the following other actions in the Microsoft Store for Business and Education portal:

- Configure Configuration Manager as the store management tool. For more information, see [Configure management provider](#).
- Enable support for offline licensed apps. For more information, see [Distribute offline apps](#).
- Acquire at least one app. For more information, see [Find and acquire apps](#).

On the **Configurations** page of the Azure Services Wizard, specify the following information:

- **Path to Microsoft Store for Business app content storage:** Specify a shared network path, including a folder. For example, `\server\share\folder`. When the site server syncs with the store, it caches content in this location. When you create an application in Configuration Manager, the site server copies the app content from this local cache to the site's content library.
- **Selected languages:** Select the languages to sync from the store and display to users in Software Center. For example, if the user configures Windows for German,

then Software Center shows German strings for the store app. This behavior requires that language to be synchronized, and to exist for the specific application.

- **Default language:** If the user's language is unavailable, select a default language to use.

 **Note**

Configuration Manager doesn't synchronize the app icon from the store. If you need an icon to display for this app in Software Center, manually add it in the app properties. For more information, see [Manually specify application information](#).

Create and deploy the app

After synchronization, create and deploy the Microsoft Store for Business and Education apps similar to any other Configuration Manager application.

1. In the **Software Library** workspace of the Configuration Manager console, expand **Application Management**, then select the **License Information for Store Apps** node.
2. Choose the app you want to deploy, then select **Create Application** in the ribbon.

The site creates a Configuration Manager application containing the Microsoft Store for Business and Education app.

Then deploy and monitor this application as you would any other Configuration Manager application. For more information, see the following articles:

- [Deploy applications](#)
- [Monitor applications from the console](#)

Manage the app

In the **Software Library** workspace, expand **Application Management**, then select the **License Information for Store Apps** node.

For each store app you manage, view the following information about the app:

- App name
- App platform
- The number of licenses for the app that you own

- The number of available licenses

After deploying online apps, any updates to that app come directly from the Microsoft Store. Furthermore, Configuration Manager doesn't check version compliance of online apps, just that Windows reports the app as installed.

When deploying offline apps to Windows devices with the Configuration Manager client, don't allow users to update applications external to Configuration Manager deployments. Control of updates to offline apps is especially important in multi-user environments such as classrooms. One option to disable the Microsoft Store is by using [group policy](#).

After the Microsoft Store for Business and Education administrator acquires an offline app, don't publish the app to users via the store. This configuration makes sure that users can't install or update online. Users only receive offline app updates via Configuration Manager.

Next steps

[Troubleshoot the Microsoft Store for Business and Education integration with Configuration Manager](#)

Create App-V virtual environments in Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

In a Microsoft Application Virtualization (App-V) virtual environment in Configuration Manager, deployed virtual applications can share the same file system and registry on client Windows PCs. Unlike standard virtual applications, these applications can share data with each other. Virtual environments are created or modified on client PCs when the application is installed or when clients next evaluate their installed applications. You can order these applications so that when multiple applications try to modify a file system or registry value, the application with the highest order takes priority.

Important

Do not rely on App-V virtual environments to provide security protection, such as from malware.

Use the following procedure to create an App-V virtual environment in Configuration Manager.

Create an App-V virtual environment

1. In the Configuration Manager console, choose **Software Library > Application Management > App-V Virtual Environments**.
2. On the **Home** tab, in the **Create** group, choose **Create Virtual Environment**.
3. In the **Create Virtual Environment** dialog box, enter the following information:
 - **Name.** Enter a unique name for the virtual environment (maximum 128 characters).
 - **Description.** (Optional) Enter a description for the virtual environment.
4. To add a new deployment type to the virtual environment, choose **Add**. You must add at least one deployment type.
5. In the **Add Applications** dialog box, specify a **Group name** (maximum 128 characters). You'll use this name to refer to the group of applications that you add

to the virtual environment.

6. Choose **Add**, select the App-V 5 applications and deployment types that you want to add to the group, and then choose **OK**.
7. In the **Add Applications** dialog box, you can select **Increase Order** or **Decrease Order** to set the application that takes priority if multiple applications attempt to modify file system or registry settings in the same virtual environment.
8. To return to the **Create Virtual Environment** dialog box, choose **OK**.
9. When you're done adding groups, choose **OK** to create the virtual environment. The new virtual environment is displayed in the **App-V Virtual Environments** node of the Configuration Manager console. You can monitor the status of your virtual environments by using the App-V Virtual Environment Status report.

 **Note**

The virtual environment is added or modified on client PCs when the application is installed or when the client next evaluates installed applications.

Import and export applications

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Use Configuration Manager to import and export applications between two hierarchies. For example, copy an application from a test environment to a production environment.

Export

1. In the Configuration Manager console, select the **Applications** node. In the Create group of the ribbon, choose **Export Application**.
2. On the **General** screen, enter a path to a new ZIP file to export into. Optionally, specify whether to export *dependencies, supersedence relationships, conditions, and virtual environments, and content for the selected applications and dependencies*. Enter any necessary administrator comments, and select **Next**.
3. Verify the application and any dependencies are listed on the **Related Objects** page and select **Next**.
4. On the Summary page, select **Next**.
5. Once the process completes, it creates the ZIP file, and you can close the wizard.

ⓘ Important

If you're going to copy this application to another environment, take both the ZIP file and the folder that accompanies it. The ZIP file must exist in the same directory as the created folder.

Import

ⓘ Note

You can only import applications from UNC paths, you can't directly import from your local disk.

1. In the Configuration Manager console, select the **Applications** node. In the Create group of the ribbon, choose **Import Application**.

2. Choose the ZIP file that you'd like to import and select **Next**.
3. The File Content window shows what happens when you import the application. Select **Next**.
4. Review the summary screen and select **Next**.
5. Close the wizard. The application is now available in the site.

 **Tip**

Starting in version 2010, when you import an object in the Configuration Manager console, it now imports to the current folder. Previously, Configuration Manager always put imported objects in the root node.

Automation

If you want to automate the import and export of applications, use the following PowerShell cmdlets:

- [Import-CMApplication](#)
- [Export-CMApplication](#)

Next steps

[Deploy applications](#)

Revise and supersede applications in Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Learn how to work with Configuration Manager application versions and how to supersede applications with a new version.

Revisions

When you make revisions to an application or a deployment type, Configuration Manager creates a new revision of the application. You can display the history of each application revision. You can also view its properties, restore a previous revision of an application, or delete an old revision.

Display the history of application revisions

1. In the Configuration Manager console, go to the **Software Library** workspace, expand **Application Management**, and select the **Applications** node. Then choose the application that you want.
2. On the **Home** tab of the ribbon, in the **Application** group, select **Revision History**. This action opens the **Application Revision History** window.

View an application revision

1. In the **Application Revision History** window, select an application revision, and then select **View**.
2. In the **Properties** dialog box, examine the properties of the selected application.

Note

This view of application properties is read-only.

Restore an application revision

1. In the Application Revision History window, select an application revision, and then select **Restore**.
2. Select **Yes** to restore the selected application revision.

Delete an application revision

1. In the Application Revision History window, select an application revision, and then select **Delete**.
2. Select **Yes** to confirm.

ⓘ Important

You can only delete the current application revision after you retire the application and it has no references.

Supersedence

Application management in Configuration Manager lets you upgrade or replace existing applications by using a supersedence relationship. When you supersede an application, you specify a new deployment type to replace the deployment type of the superseded application. You can also decide whether to upgrade or uninstall the superseded application before the client installs the superseding application. It's best to limit supersedence chains to five levels deep at a maximum.

ⓘ Important

When you choose the option to uninstall the superseded deployment type, a deployment type can't be superseded by a deployment type that was deployed to a different type of collection. For example, a deployment type that was deployed to a device collection can't be superseded by a deployment type that was deployed to a user collection.

Decide whether to upgrade or replace an application

The type of supersedence depends on whether you select the **Uninstall** option:

- If you want to update to a newer version of the same application with the same application ID, *don't* select **Uninstall**.

- If you want to change to a different application with a different application ID, select **Uninstall**. You need to remove the superseded version of the application.

Supersede dependent applications

In this example, *main application* refers to the app that you're deploying that has the dependencies.

You can create a supersedence relationship that updates the dependent application to a new version.

1. Make sure that the new dependent application and the original dependent application are in the same dependency group of the main application.
2. Create a supersedence relationship that supersedes the original dependent application with the new dependent application.

During new installations of the main application, the client installs the new dependent application. Configuration Manager updates existing installations of the main application with the new dependent application.

The end result is that all deployments of the main application use the new dependent application.

Further considerations

- You can specify multiple supersedence relationships for dependent applications. Configuration Manager installs the highest dependent application in the supersedence chain.
- Deploy dependent applications to the device where the main application is installed. Otherwise Configuration Manager won't install the dependent application.
- For new installations of the main application, when you have multiple dependencies, the dependency order determines which version of the dependent application gets installed.

Specify a supersedence relationship

1. In the Configuration Manager console, go to the **Software Library** workspace, expand **Application Management**, and select the **Applications** node. Then choose the application that supersedes another application.

2. On the **Home** tab of the ribbon, in the **Properties** group, select **Properties**.
3. Switch to the **Supersedence** tab, and select **Add**.
4. For the **Superseded Application**, select **Browse**.
5. Choose the application that you want to supersede, and then select **OK**.
6. In the **Specify Supersedence Relationship** window, select the deployment type that replaces the deployment type of the superseded application.

 **Note**

By default, the new deployment type doesn't uninstall the deployment type of the superseded application. This scenario is commonly used when you want to deploy an upgrade to an existing application. To remove the existing deployment type before the new deployment type is installed, select **Uninstall**. If you decide to upgrade an application, make sure that you test this in a lab environment first.

7. If you want users to still see Software Center deployments for both applications, select the option to **Allow users to see deployments for this application and all applications that it supersedes in Software Center**. With this option, you give users the choice to still install an older version of the app if needed. By default, this option isn't selected, so only the superseding application displays in Software Center. This option is only for available deployments to user collections.
8. Select **OK** to save your changes and close the windows.

Display applications that supersede the current application

1. In the Configuration Manager console, go to the **Software Library** workspace, expand **Application Management**, and select the **Applications** node. Then choose the application that you want.
2. On the **Home** tab of the ribbon, in the **Properties** group, select **Properties**.
3. Switch to the **References** tab.
4. For the **Relationship type**, choose **Applications that supersede this application**.

View supersedence relationships

1. In the Configuration Manager console, go to the **Software Library** workspace, expand **Application Management**, and select the **Applications** node. Then choose the application that you want.
2. On the **Home** tab of the ribbon, in the **Relationships** group, select **View relationships**, and then select **Supersedence**.

This action shows a graphical diagram of the relationships of the selected application to other applications. For the supersedence relationships, it shows applications that the selected application supersedes, and applications that the selected application is superseded by.

Manage supersedence with PowerShell

You can add, view, and remove supersedence relationships using the following PowerShell cmdlets:

- [Get-CMDeploymentTypeSupersedence](#)
- [Set-CMAApplicationSupersedence](#)

Next steps

[Uninstall applications](#)

Uninstall applications with Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Instead of needing to create a separate object to uninstall an application, you can specify uninstall behaviors on the deployment type. Then create a separate deployment with the action to uninstall. You can uninstall an application even if it wasn't previously installed by Configuration Manager.

Behaviors and limitations

- To deploy an application with the **Uninstall** action, first delete any existing application deployments, simulated deployments, or task sequence deployments that include this application. Otherwise Configuration Manager may reinstall the application.
- Some application types don't support uninstallation.
- When you uninstall an application, Configuration Manager doesn't automatically uninstall dependencies.
- If you deploy to a user an application with the **Uninstall** action, and the application was installed for all users of the computer, the uninstall might fail if the user's account doesn't have permissions to uninstall the application.
- In version 2103 and earlier, if you remove a user or a device from a collection that has an application deployed to it, Configuration Manager doesn't automatically uninstall the application from the device.

Tip

Version 2107 and later supports **Implicit uninstall**.

- A deployment with the **Uninstall** action doesn't check requirement rules. If the application is installed on the target device, Configuration Manager uninstalls it.

Process

When you [create the application](#), select the option to **Automatically identify information about this deployment type from installation files**. If the information is available in the installation files, the uninstall command line is automatically added to the deployment type properties.

For an existing application, use the following steps to configure its uninstall properties:

1. In the Configuration Manager console, go to the **Software Library** workspace. Expand **Application Management** and select the **Applications** node.
2. Select the application. In the details pane, switch to the **Deployment Types** tab.
3. Select the deployment type. Then in the ribbon, on the **Deployment Type** tab, select **Properties**.
4. Switch to the **Content** tab and configure the following settings:
 - **Uninstall content settings:** Select an option for where Configuration Manager gets the content to uninstall the application:
 - **Same as install content:** The install and uninstall content are the same. This option is the default.
 - **No uninstall content:** Your application doesn't need content for uninstall.
 - **Different from install content:** The uninstall content is different from the install content.
 - **Uninstall content location:** If you select the third option for content settings, specify the network path to the content that's used to uninstall the application.
5. Switch to the **Programs** tab and configure the following settings:
 - **Uninstall program:** Specify the command line and any required parameters to uninstall the application.
 - **Uninstall start in:** Optionally specify the folder that has the uninstall program for the deployment type. This folder can be an absolute path on the client. It can also be a relative path on a distribution point of the folder with the package.
 - **Run installation and uninstall program as 32-bit process on 64-bit clients:** Use the 32-bit file and registry locations on Windows-based computers to run the uninstall program for the deployment type.

Then [deploy the application](#). On the **Deployment Settings** page of the wizard, select the deployment action to **Uninstall**.

① Note

When you select a deployment action of **Uninstall**, the deployment purpose is automatically configured as **Required**.

Implicit uninstall

Many customers have lots of collections because for every application they need at least two collections: one for install and another for uninstall. This practice adds overhead of managing more collections, and can reduce site performance for collection evaluation.

Starting in version 2107, you can enable an application deployment to support implicit uninstall. If a resource is in a collection, the application installs. Then when you remove the resource from the collection, the application uninstalls.

Starting in version 2111, this behavior also supports [application groups](#). When this article refers to an *application*, it also applies to app groups.

① Note

In version 2111 and later, this behavior applies to deployments to device or user collections. In version 2107, this behavior only applies to deployments to device collections.

Starting in version 2203, if you deploy an application or app group to a user collection that's based on a security group, and you enable implicit uninstall, changes to the security group are now honored. When the site discovers the change in group membership, Configuration Manager uninstalls the app for the user that you removed from the security group.

Enable implicit uninstall

When you [deploy the application](#) to a collection, configure the following settings on the **Deployment Settings** page:

- **Action:** Install
- **Purpose:** Required

- Enable the following option: **When a resource is no longer a member of the collection, uninstall the application**

 **Tip**

In version 2107, this option is named: **Uninstall this application if the targeted object falls out of the collection**

 **Important**

Be careful with enabling this option on deployments to large query-based collections. Especially queries to external sources like Active Directory groups. An unexpected external change could automatically trigger a large number of devices to uninstall the application.

Implicit uninstall process

After you remove the resource from the collection, the following process happens:

- A background worker process runs on the site server every 10 minutes. This task keeps track of apps for which you've enabled this option. It then detects resources that you removed from the target collection. To help you troubleshoot this process, view the **SMS_ImplicitUninstall.log** file on the site server.
- The client needs to download policy. By default, the [client policy polling interval](#) client setting is 60 minutes. To accelerate this step, manually [download policy](#).
- 15 minutes after the client receives the updated policy, it uninstalls the app.

Depending upon the timing of those steps, the longest time period for the client to uninstall the app is 85 minutes. If the first step happens immediately, and you manually download policy on the device, the overall process is 15 minutes.

 **Note**

- For this behavior, the site can process up to 1000 collection membership changes every 10 minutes.
- If the uninstall doesn't occur, it's likely that there's a conflicting install deployment of the same application, application group, or a different

application group with the same apps. Configuration Manager always honors an install deployment over an uninstall deployment.

Known issues

You configure an app's installation behavior to **Install for system**, and then deploy it to a user collection. A device has multiple users who are both in the collection, and the app installs on the device. If you then remove *one user* from the collection, the app is uninstalled from the device for all users.

Next steps

[How to manage collections](#)

[Monitor applications from the Configuration Manager console](#)

[Log file reference](#)

Create and run PowerShell scripts from the Configuration Manager console

Article • 10/09/2023

Applies to: Configuration Manager (current branch)

Configuration Manager has an integrated ability to run PowerShell scripts. PowerShell has the benefit of creating sophisticated, automated scripts that are understood and shared with a larger community. The scripts simplify building custom tools to administer software and let you accomplish mundane tasks quickly, allowing you to get large jobs done more easily and more consistently.

Note

In version 2006 and earlier, Configuration Manager doesn't enable this optional feature by default. You must enable this feature before using it. For more information, see [Enable optional features from updates](#).

With this integration in Configuration Manager, you can use the *Run Scripts* functionality to do the following things:

- Create and edit scripts for use with Configuration Manager.
- Manage script usage through roles and security scopes.
- Run scripts on collections or individual on-premises managed Windows PCs.
- Schedule scripts' runtime in UTC on collections or individual on-premises managed Windows PCs.
- Get rapid aggregated script results from client devices.
- Monitor script execution and view reporting results from script output.

Warning

- Given the power of scripts, we remind you to be intentional and careful with their usage. We have built in additional safeguards to assist you; segregated roles and scopes. Be sure to validate the accuracy of scripts before running them and confirm they are from a trusted source, to prevent unintended script execution. Be mindful of extended characters or other obfuscation and educate yourself about securing scripts. [Learn more about PowerShell script security](#)

- Certain anti-malware software may inadvertently trigger events against the Configuration Manager Run Scripts or CMPivot features. It is recommended to exclude %windir%\CCM\ScriptStore so that the anti-malware software permits those features to run without interference.

Prerequisites

- To run PowerShell scripts, the client must be running PowerShell version 3.0 or later. However, if a script you run contains functionality from a later version of PowerShell, the client on which you run the script must be running that version of PowerShell.
- Configuration Manager clients must be running the client from the 1706 release, or later in order to run scripts.
- To use scripts, you must be a member of the appropriate Configuration Manager security role.
- To import and author scripts - Your account must have **Create** permissions for **SMS Scripts**.
- To approve or deny scripts - Your account must have **Approve** permissions for **SMS Scripts**.
- To run scripts - Your account must have **Run Script** permissions for **Collections**.

For more information about Configuration Manager security roles:

[Security scopes for run scripts](#)

[Security roles for run scripts](#)

[Fundamentals of role-based administration](#).

Limitations

Run Scripts currently supports:

- Scripting languages: PowerShell
- Parameter types: integer, string, and list.

Warning

Be aware that when using parameters, it opens a surface area for potential PowerShell injection attack risk. There are various ways to mitigate and work around, such as using regular expressions to validate parameter input or using predefined parameters. Common best practice is not to include secrets in your

PowerShell scripts (no passwords, etc.). [Learn more about PowerShell script security](#)

Run Script authors and approvers

Run Scripts uses the concept of *script authors* and *script approvers* as separate roles for implementation and execution of a script. Having the author and approver roles separated allows an important process check for the powerful tool that Run Scripts is. There's an additional *script runners* role that allows execution of scripts, but not creation or approval of scripts. See [Create security roles for scripts](#).

Scripts roles control

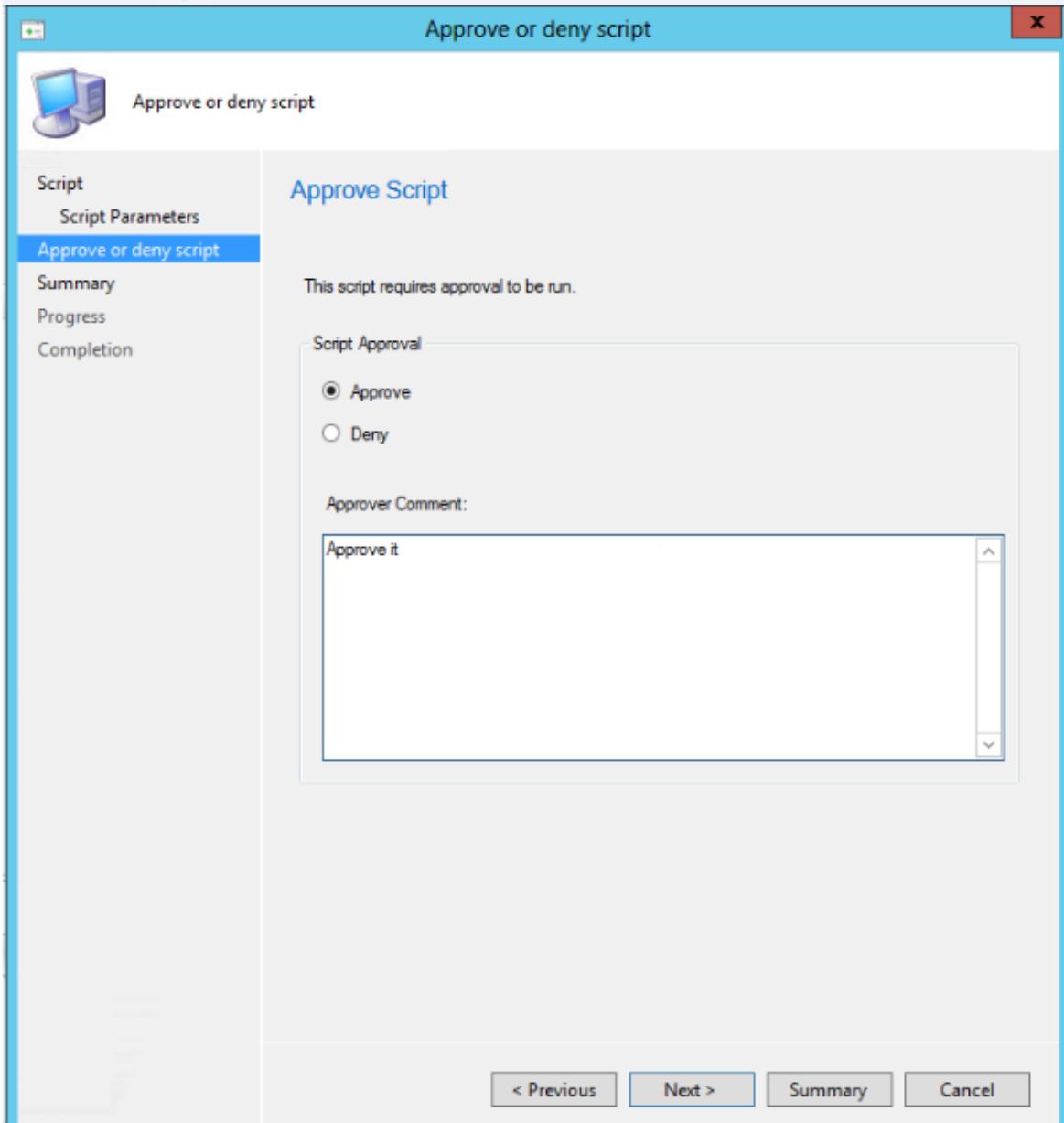
By default, users can't approve a script they've authored. Because scripts are powerful, versatile, and potentially deployed to many devices, you can separate the roles between the person that authors the script and the person that approves the script. These roles give an additional level of security against running a script without oversight. You're able to turn off secondary approval, for ease of testing.

Approve or Deny a script

Scripts must be approved, by the *script approver* role, before they can be run. To approve a script:

1. In the Configuration Manager console, click **Software Library**.
2. In the **Software Library** workspace, click **Scripts**.
3. In the **Script** list, choose the script you want to approve or deny and then, on the **Home** tab, in the **Script** group, click **Approve/Deny**.
4. In the **Approve or deny script** dialog box, select **Approve**, or **Deny** for the script. Optionally, enter a comment about your decision. If you deny a script, it can't be

run on client devices.



5. Complete the wizard. In the **Script** list, you see the **Approval State** column change depending on the action you took.

Allow users to approve their own scripts

This approval is primarily used for the testing phase of script development.

1. In the Configuration Manager console, click **Administration**.
2. In the **Administration** workspace, expand **Site Configuration**, and then click **Sites**.
3. In the list of sites, choose your site and then, on the **Home** tab, in the **Sites** group, click **Hierarchy Settings**.
4. On the **General** tab of the **Hierarchy Settings Properties** dialog box, clear the checkbox **Script authors require additional script approver**.

ⓘ Important

As a best practice, you shouldn't allow a script author to approve their own scripts. It should only be allowed in a lab setting. Carefully consider the potential impact of changing this setting in a production environment.

Security scopes

Run Scripts uses security scopes, an existing feature of Configuration Manager, to control scripts authoring and execution through assigning tags that represent user groups. For more information on using security scopes, see [Configure role-based administration for Configuration Manager](#).

Create security roles for scripts

The three security roles used for running scripts aren't created by default in Configuration Manager. To create the script runners, script authors, and script approvers roles, follow the outlined steps.

1. In the Configuration Manager console, go to **Administration > Security > Security Roles**
2. Right-click on a role and click **Copy**. The role you copy has permissions already assigned. Make sure you take only the permissions that you want.
3. Give the custom role a **Name** and a **Description**.
4. Assign the security role the permissions outlined below.

Security Role Permissions

Role Name: Script Runners

- **Description:** These permissions enable this role to only run scripts that were previously created and approved by other roles.
- **Permissions:** Ensure the following are set to **Yes**.

Category	Permission	State
Collection	Run Script	Yes
Site	Read	Yes
SMS Scripts	Read	Yes

Role Name: Script Authors

- **Description:** These permissions enable this role to author scripts, but they can't approve or run them.
- **Permissions:** Ensure the following permissions are set.

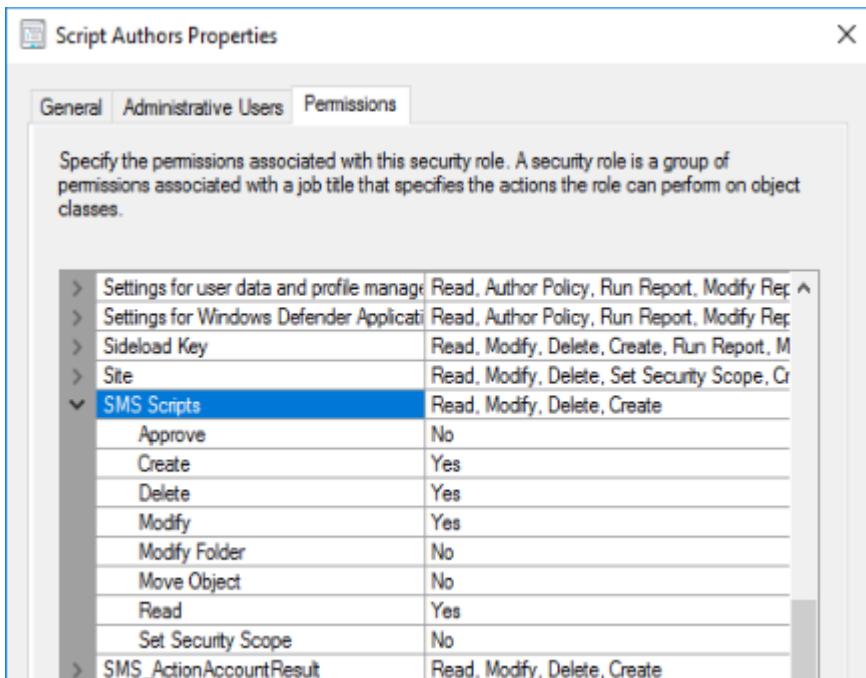
Category	Permission	State
Collection	Run Script	No
Site	Read	Yes
SMS Scripts	Create	Yes
SMS Scripts	Read	Yes
SMS Scripts	Delete	Yes
SMS Scripts	Modify	Yes

Role Name: Script Approvers

- **Description:** These permissions enable this role to approve scripts, but they can't create or run them.
- **Permissions:** Ensure the following permissions are set.

Category	Permission	State
Collection	Run Script	No
Site	Read	Yes
SMS Scripts	Read	Yes
SMS Scripts	Approve	Yes
SMS Scripts	Modify	Yes

Example of SMS Scripts permissions for the script authors role



Create a script

1. In the Configuration Manager console, click **Software Library**.
2. In the **Software Library** workspace, click **Scripts**.
3. On the **Home** tab, in the **Create** group, click **Create Script**.
4. On the **Script** page of the **Create Script** wizard, configure the following settings:
 - **Script Name** - Enter a name for the script. Although you can create multiple scripts with the same name, using duplicate names makes it harder for you to find the script you need in the Configuration Manager console.
 - **Script language** - Currently, only PowerShell scripts are supported.
 - **Import** - Import a PowerShell script into the console. The script is displayed in the **Script** field.
 - **Clear** - Removes the current script from the **Script** field.
 - **Script** - Displays the currently imported script. You can edit the script in this field as necessary.
5. Complete the wizard. The new script is displayed in the **Script** list with a status of **Waiting for approval**. Before you can run this script on client devices, you must approve it.

ⓘ Important

Avoid scripting a device reboot or a restart of the Configuration Manager agent when using the Run Scripts feature. Doing so could lead to a continuous rebooting state. If needed, there are enhancements to the client notification feature that

enable restarting devices. The **pending restart column** can help identify devices that need a restart.

Script parameters

Adding parameters to a script provides increased flexibility for your work. You can include up to 10 parameters. The following outlines the Run Scripts feature's current capability with script parameters for; *String*, *Integer* data types. Lists of preset values are also available. If your script has unsupported data types, you get a warning.

In the **Create Script** dialog, click **Script Parameters** under **Script**.

Each of your script's parameters has its own dialog for adding further details and validation. If there's a default parameter in the script, it will be enumerated in the parameter UI and you can set it. Configuration Manager won't overwrite the default value since it will never modify the script directly. You can think of this as "pre-populated suggested values" are provided in the UI, but Configuration Manager doesn't provide access to "default" values at run-time. This can be worked around by editing the script to have the correct defaults.

Important

Parameter values can't contain a single quote.

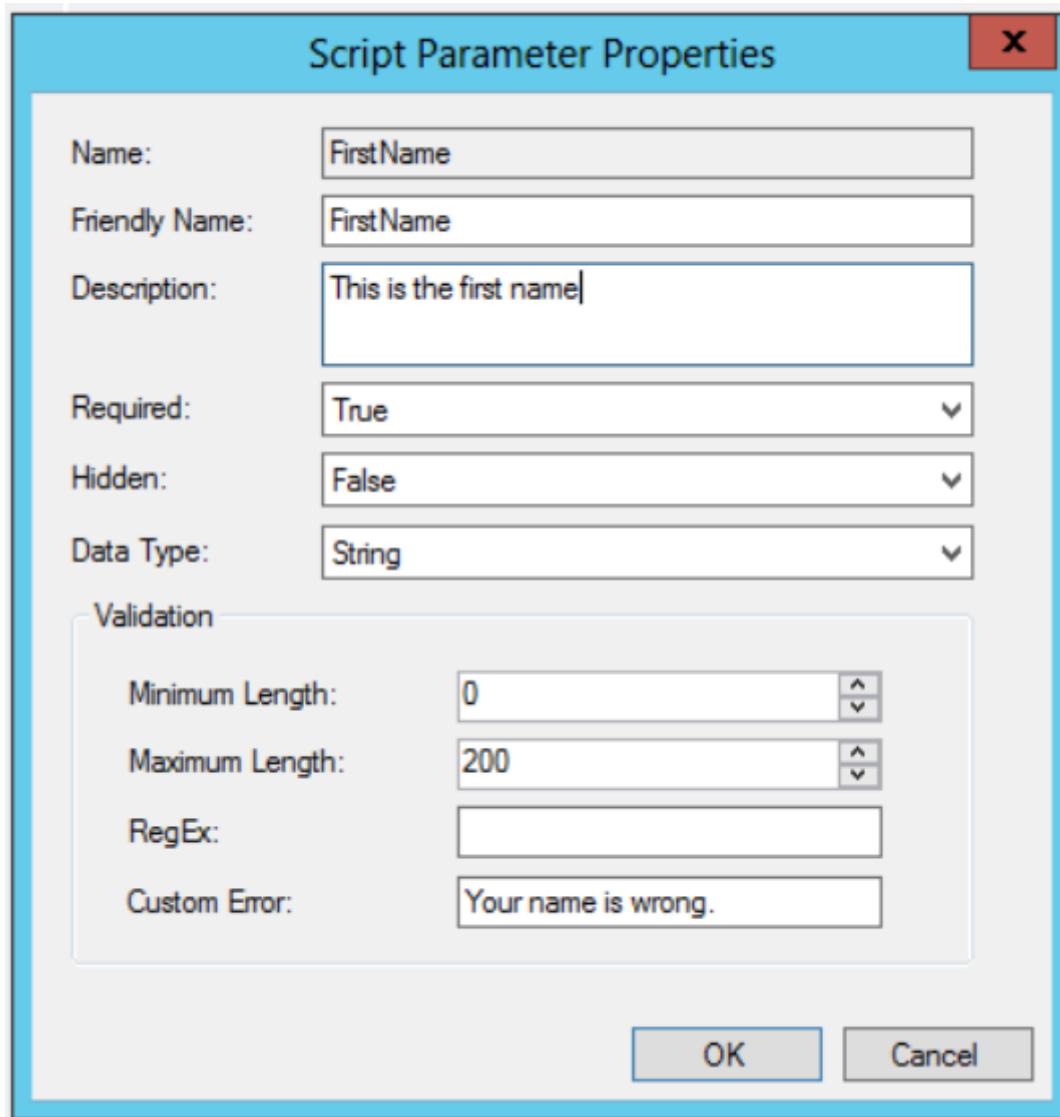
There is a known issue where parameter values that include or are enclosed in single quotes don't get passed to the script properly. When specifying default parameter values containing a space within a script, use double quotes instead. When specifying default parameter values during creation or execution of a **Script**, surrounding the default value in either double or single quotes is not necessary regardless of whether the value contains a space or not.

Parameter validation

Each parameter in your script has a **Script Parameter Properties** dialog for you to add validation for that parameter. After adding validation, you should get errors if you're entering a value for a parameter that doesn't meet its validation.

Example: *FirstName*

In this example, you're able to set the properties of the string parameter, *FirstName*.



The validation section of the **Script Parameter Properties** dialog contains the following fields for your use:

- **Minimum Length** - minimum number of characters of the *FirstName* field.
- **Maximum Length** - maximum number of characters of the *FirstName* field
- **RegEx** - short for *Regular Expression*. For more information on using the Regular Expression, see the next section, *Using Regular Expression validation*.
- **Custom Error** - useful for adding your own custom error message that supersedes any system validation error messages.

Using Regular Expression validation

A regular expression is a compact form of programming for checking a string of characters against an encoded validation. For example, you could check for the absence of a capital alphabetic character in the *FirstName* field by placing `[^A-Z]` in the *RegEx* field.

The regular expression processing for this dialog is supported by the .NET Framework. For guidance on using regular expressions, see [.NET Regular Expression](#) and [Regular Expression Language](#).

Script examples

Here are a couple examples that illustrate scripts you might want to use with this capability.

Create a new folder and file

This script creates a new folder and a file within the folder, given your naming input.

```
PowerShell

Param(
[Parameter(Mandatory=$True)]
[string]$FolderName,
[Parameter(Mandatory=$True)]
[string]$FileName
)

New-Item $FolderName -type directory
New-Item $FileName -type file
```

Get OS Version

This script uses WMI to query the machine for its OS version.

```
PowerShell

Write-Output (Get-WmiObject -Class Win32_operatingSystem).Caption
```

Edit or copy PowerShell scripts

You can **Edit** or **Copy** an existing PowerShell script used with the **Run Scripts** feature. Instead of recreating a script that you need to change, now directly edit it. Both actions use the same wizard experience as when you create a new script. When you edit or copy a script, Configuration Manager doesn't persist the approval state.

 **Tip**

Don't edit a script that's actively running on clients. They won't finish running the original script, and you may not get the intended results from these clients.

Edit a script

1. Go to the **Scripts** node under the **Software Library** workspace.
2. Select the script to edit, then click **Edit** in the ribbon.
3. Change or reimport your script in the **Script Details** page.
4. Click **Next** to view the **Summary** then **Close** when you're finished editing.

Copy a script

1. Go to the **Scripts** node under the **Software Library** workspace.
2. Select the script to copy, then click **Copy** in the ribbon.
3. Rename the script in the **Script name** field and make any additional edits you may need.
4. Click **Next** to view the **Summary** then **Close** when you're finished editing.

Run a script

After a script is approved, it can be run against a single device or a collection. Once execution of your script begins, it's launched quickly through a high priority system that times out in one hour. The results of the script are then returned using a state message system.

To select a collection of targets for your script:

1. In the Configuration Manager console, click **Assets and Compliance**.
2. In the Assets and Compliance workspace, click **Device Collections**.
3. In the **Device Collections** list, click the collection of devices on which you want to run the script.
4. Select a collection of your choice, click **Run Script**.
5. On the **Script** page of the **Run Script** wizard, choose a script from the list. Only approved scripts are shown.
6. Click **Next**, and then complete the wizard.

Important

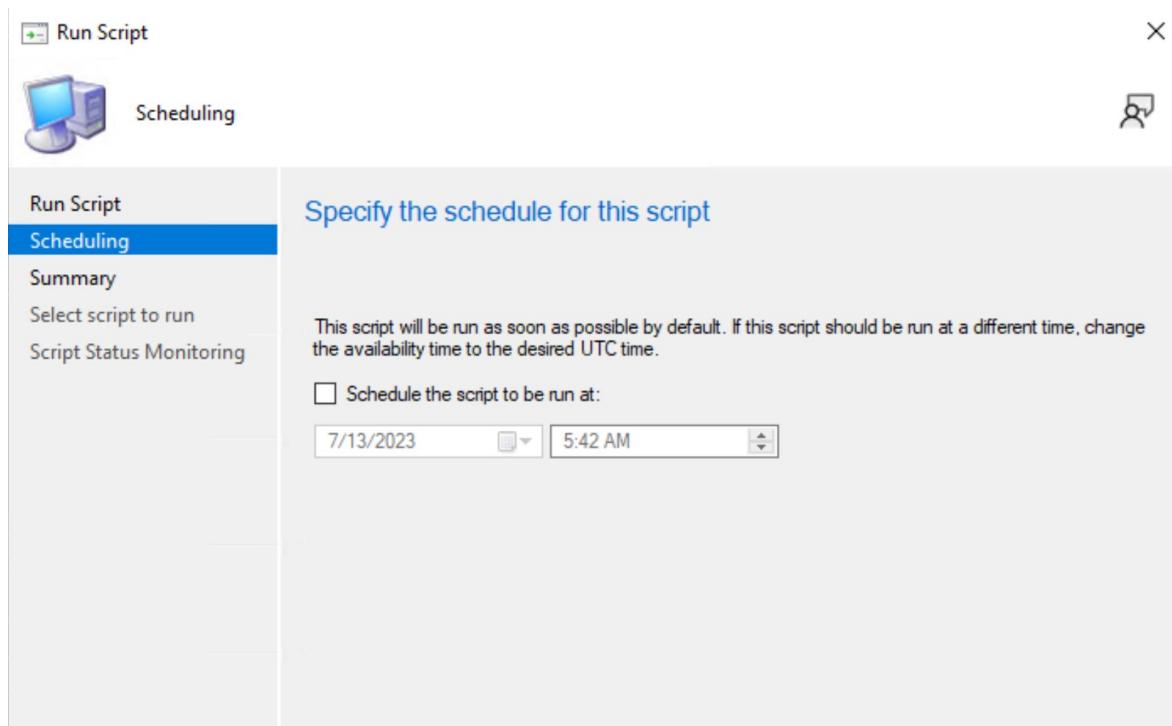
If a script does not run, for example because a target device is turned off during the one hour time period, you must run it again.

Schedule scripts' runtime

Starting in Configuration Manager current branch version 2309, you can now schedule scripts' runtime in UTC.

Schedule script execution on a collection:

1. In the Configuration Manager console, click **Assets and Compliance**.
2. In the Assets and Compliance workspace, click **Device Collections**.
3. In the **Device Collections** list, click the collection of devices on which you want to schedule the script.
4. Select a collection of your choice, click **Run Script**.
5. On the **Scheduling page**, Schedule the script to be run at checkbox and specify the Schedule Time in UTC.
6. Verify the details that are displayed on the **summary page**.
7. Click **Next**, and then complete the wizard.



ⓘ Note

A max of twenty five scheduled scripts will be processed in every 5 minutes.

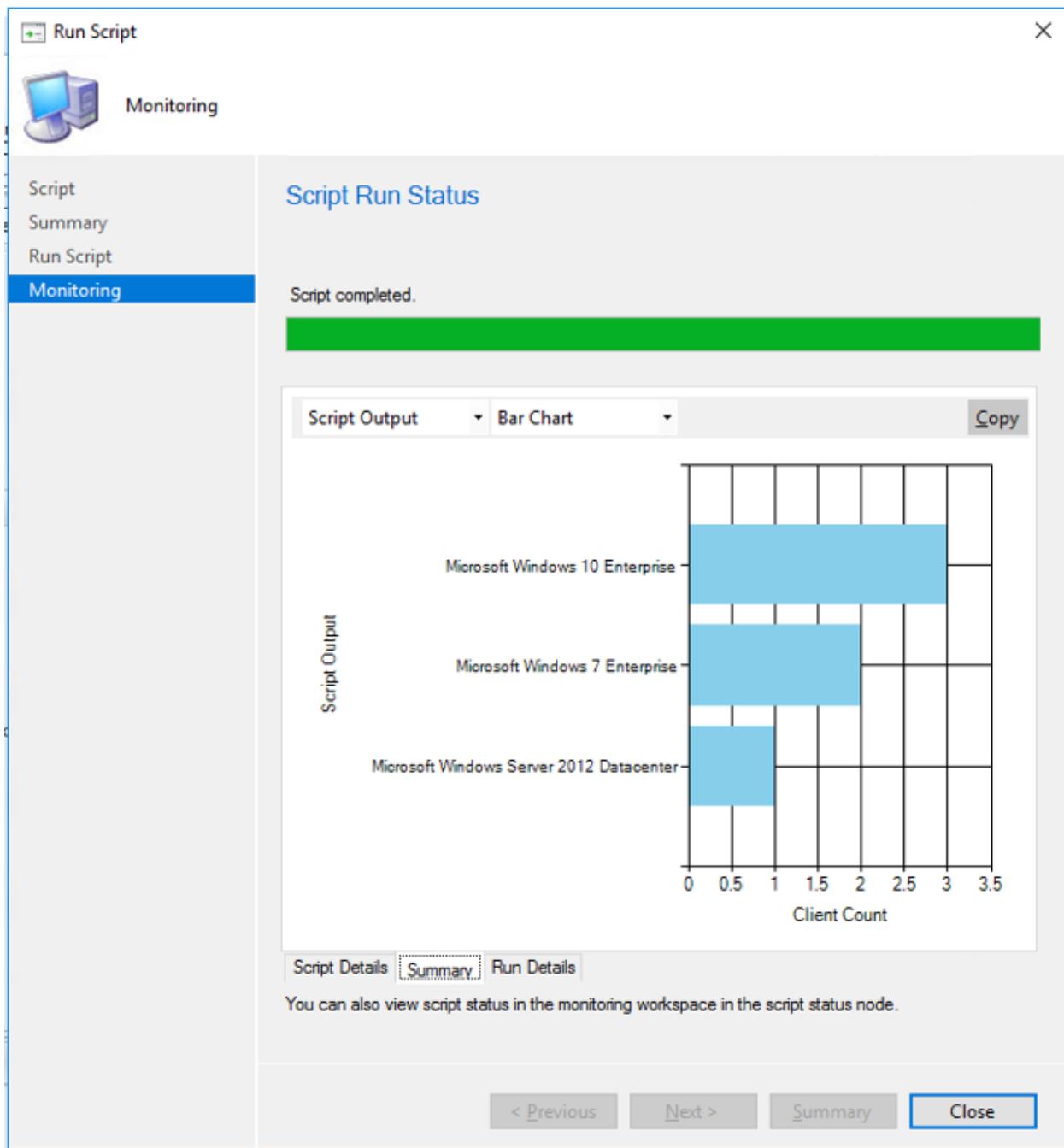
Target machine execution

The script is executed as the *system* or *computer* account on the targeted client(s). This account has limited network access. Any access to remote systems and locations by the script must be provisioned accordingly.

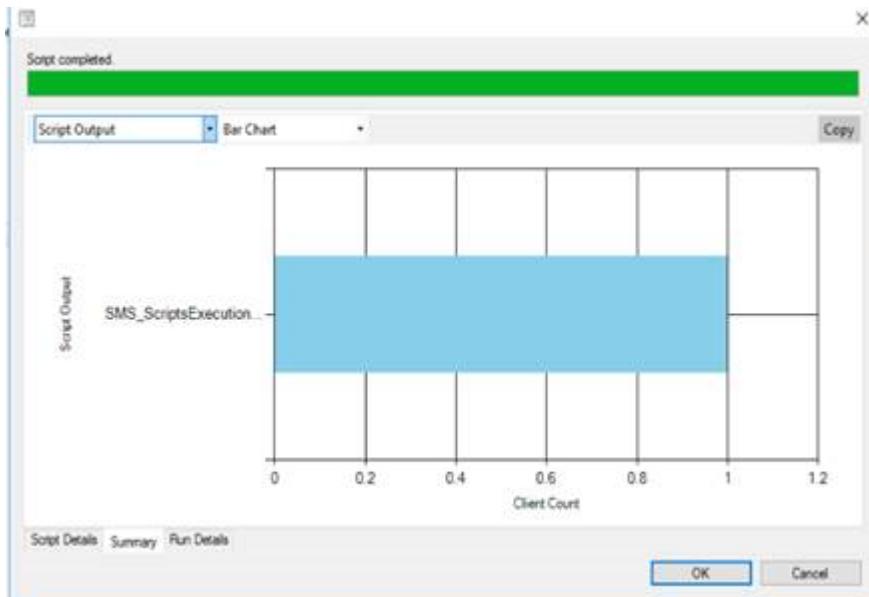
Script monitoring

After you have initiated running a script on a collection of devices, use the following procedure to monitor the operation. You are able to monitor a script in real time as it executes, and later return to the status and results for a given Run Script execution.

Script status data is cleaned up as part of the [Delete Aged Client Operations maintenance task](#) or deletion of the script.

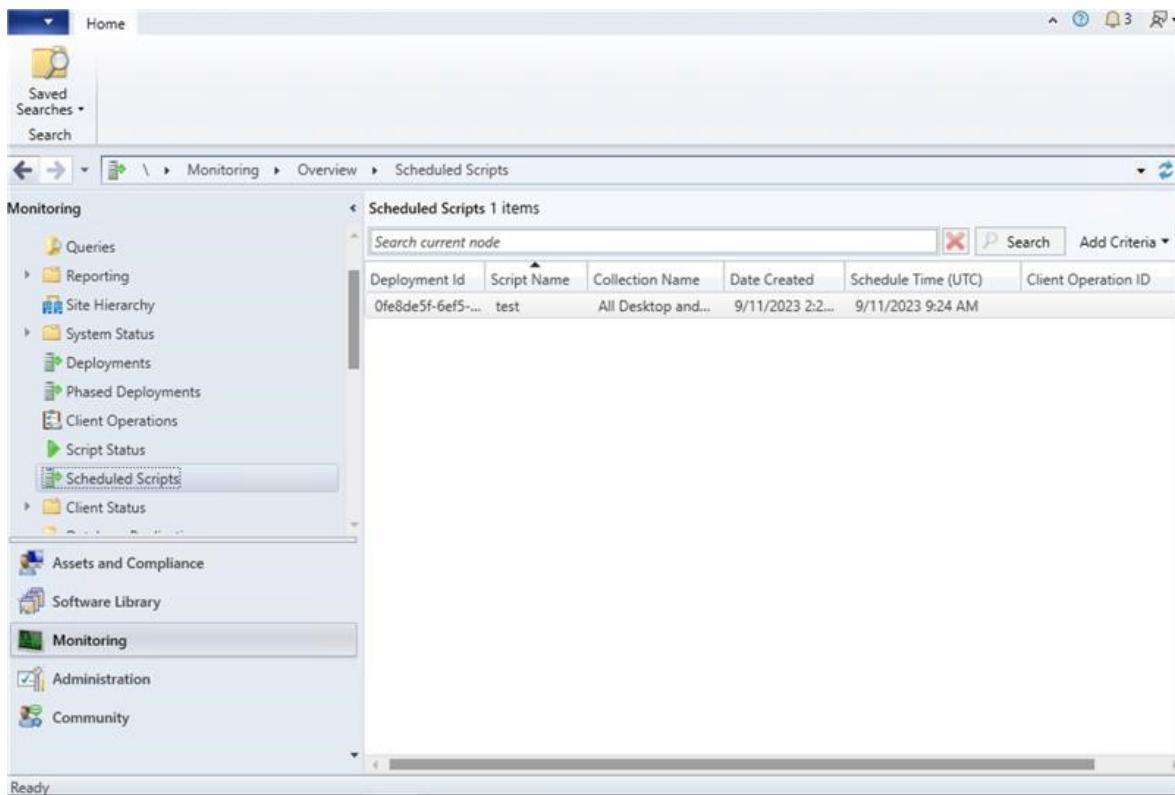


1. In the Configuration Manager console, click **Monitoring**.
2. In the **Monitoring** workspace, click **Script Status**.
3. In the **Script Status** list, you view the results for each script you ran on client devices. A script exit code of 0 generally indicates that the script ran successfully.



Schedule script Monitoring on a collection

1. In the Configuration Manager console, click **Monitoring**.
2. In the Monitoring workspace, click **Scheduled Scripts node**.
3. A new row will be displayed in the list of **Scheduled Scripts**.
4. Verify a new row has been displayed in the list of Scheduled Scripts. The state column should have the value **Scheduled**. The **ClientOperationId** column should be blank. Verify that the other columns like Script Name, Schedule Time etc. have appropriate values.
5. After the Schedule Time, refresh the **Scheduled Scripts** node. The state column should have the value **Successfully initiated client operation**. The **ClientOperationId** column should have an integer value.
6. In the Monitoring workspace, click **Script Status node**. Verify new row has been displayed in the list and the **ClientOperationId** is equal to the ClientOperationId from the **Scheduled Scripts** node.
7. Click on **View Status** and ensure that the script output displays.



Script output

Client's return script output using JSON formatting by piping the script's results to the [ConvertTo-Json](#) cmdlet. The JSON format consistently returns readable script output. For scripts that do not return objects as output, the ConvertTo-Json cmdlet converts the output to a simple string that the client returns instead of JSON.

- Scripts that get an unknown result, or where the client was offline, won't show in the charts or data set.
- Avoid returning large script output since it's truncated to 4 KB.
- Convert an enum object to a string value in scripts so they're properly displayed in JSON formatting.

```
PS C:\windows\system32> Get-ExecutionPolicy |ConvertTo-Json  
1  
  
PS C:\windows\system32> (Get-ExecutionPolicy).ToString() |ConvertTo-Json  
"RemoteSigned"
```

You can view detailed script output in raw or structured JSON format. This formatting makes the output easier to read and analyze. If the script returns valid JSON-formatted text or the output can be converted to JSON using the [ConvertTo-Json](#) PowerShell cmdlet, then view the detailed output as either **JSON Output** or **Raw Output**. Otherwise the only option is **Script Output**.

Example: Script output is convertible to valid JSON

Command: `$PSVersionTable.PSVersion`

```
Output

Major   Minor   Build   Revision
-----  -----  -----  -----
5        1       16299  551
```

Example: Script output isn't valid JSON

Command: `Write-Output (Get-WmiObject -Class Win32_OperatingSystem).Caption`

```
Output

Microsoft Windows 10 Enterprise
```

Log files

- On the client, by default in C:\Windows\CCM\logs:
 - `Scripts.log`
 - `CcmMessaging.log`
- On the MP, by default in C:\SMS_CCM\Logs:
 - `MP_RelayMsgMgr.log`
- On the site server, by default in C:\Program Files\Configuration Manager\Logs:
 - `SMS_Message_Processing_Engine.log`

Automate with Windows PowerShell

You can use the following PowerShell cmdlets to automate some of these tasks:

- [Approve-CMScript](#)
- [Deny-CMScript](#)
- [Get-CMScript](#)
- [Invoke-CMScript](#)
- [New-CMScript](#)
- [Remove-CMScript](#)
- [Set-CMScript](#)

See also

- [Configure role-based administration for Configuration Manager](#)
- [Fundamentals of role-based administration](#)

Learn more about PowerShell script security

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

It's the administrator's responsibility to validate proposed PowerShell and PowerShell parameter usage in their environment. Here are some helpful resources to help educate administrators about the power of PowerShell and potential risk surfaces. This guidance is to help you mitigate potential risk surfaces and allow safe scripts to be used.

PowerShell Script Security

The Configuration Manager scripts feature lets you visually review and approve scripts. Another administrator can request that their script is allowed. Administrators should be aware PowerShell scripts can have obfuscated scripts. An obfuscated script could be malicious and difficult to detect with visual inspection during the script approval process. Visually review PowerShell scripts and use inspection tools to help detect suspicious script issues. These tools can't always determine the PowerShell author's intent, so it can bring attention to a suspicious script. However, the tools will require the administrator to judge if it's malicious or intentional script syntax.

Recommendations

- Familiarize yourself with PowerShell security guidance using the various links referenced below.
- **Sign your scripts:** Another method for keeping scripts secure is by having them vetted and then signed, before importing them for usage.
- Don't store secrets (such as passwords) in PowerShell scripts and learn more about how to handle secrets.

General information about PowerShell security

This collection of links was chosen to give Configuration Manager administrators a starting point for learning about PowerShell script security recommendations.

[Defending Against PowerShell Attacks](#)

[Protecting Against Malicious Code Injection](#)

API for anti-malware scan interface ↗

PowerShell parameters security

Passing parameters is a way to have flexibility with your scripts and defer decisions until run time. It also opens up another risk surface.

The following list includes recommendations to prevent malicious parameters or script injection:

- Only allow usage of pre-defined parameters.
- Use the regular expression feature, to validate parameters that are allowed.
 - Example: If only a certain range of values are allowed, use a regular expression to check for only those characters or values that can make up the range.
 - Validating parameters can help prevent users trying use certain characters that can be escaped, like quotes. There can be multiple types of quotes, so using regular expressions to validate which characters you've decided are permissible is often easier than trying to define all the inputs that not permissible.
- Use the PowerShell module "[injection hunter](#)" ↗ in the PowerShell Gallery.
 - There can be false positives, so look for intent when something is flagged as suspicious to determine if it's a real issue or not.
- Microsoft Visual Studio has a script analyzer, that can assist with checking PowerShell syntax.

The following video titled: "DEF CON 25 - Lee Holmes - Get \$pwnd: Attacking Battle Hardened Windows Server" gives an overview of the types of issues that you can secure against (especially the section 12:20 to 17:50):

<https://www.youtube-nocookie.com/embed/ahxMOAAani8> ↗

Environment recommendations

The following list includes general recommendations for PowerShell administrators:

- Deploy the latest version of PowerShell, such as version 5 or later, which is built into Windows 10 or later. You can also deploy the [Windows Management Framework](#) ↗ .
- Enable, and collect PowerShell logs, optionally including Protected Event Logging. Incorporate these logs into your signatures, hunting, and incident response workflows.

- Implement Just Enough Administration on high-value systems to eliminate or reduce unconstrained administrative access to those systems.
- Deploy Windows Defender Application Control policies to allow pre-approved administrative tasks to use the full capability of the PowerShell language, while limiting interactive and unapproved use to a limited subset of the PowerShell language.
- Deploy Windows 10 or later to give your antivirus provider full access to all content (including content generated or de-obfuscated at runtime) processed by Windows Scripting Hosts including PowerShell.

Package Conversion Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Package Conversion Manager helps you convert Configuration Manager legacy packages into applications. Applications have additional benefits such as dependencies, requirement rules, detection methods, and user device affinity.

A Configuration Manager application contains files and programs that you deploy to client devices. However, unlike legacy packages and programs, an application provides additional user-centric functionality. For example, an application might contain deployment types for a local installation of a software package, a virtual application package, or a version of the application for mobile devices.

For more information, see the following articles:

- [Introduction to application management](#)
- [Packages and programs](#)

Important

If you previously installed an older version of Package Conversion Manager, first uninstall it before upgrading your site. This integrated version doesn't require installation, but may conflict with existing versions.

This integrated version of Package Conversion Manager works on packages in the Configuration Manager current branch site. It's not a standalone tool. If you have packages and programs in an older version of Configuration Manager, first migrate the packages into your current branch site. For more information, see [Migrate data between hierarchies](#).

Planning

Before you start converting packages into applications, first develop a plan. The following process is an example plan:

- [Define a detailed package conversion plan](#)
- [Select and prepare packages for conversion](#)
- [Select test packages](#)

- Analyze, investigate, and convert packages
- Test and deploy the applications

Define a detailed package conversion plan

This section describes two sample package conversion plans:

- [A high-resource test environment](#): You have a test environment with the resources, permissions, and architecture to fully replicate your production environment.
- [A limited-resource test environment](#): You don't have a test environment that fully replicates your production environment.

Adjust these plans as necessary for other issues specific to your environment.

Sample plan for a high-resource test environment

Your test environment has the resources, permissions, and architecture similar to your production environment. Use the test environment to efficiently analyze and convert all of your packages, and then test all of your Configuration Manager applications. After completing that work, transfer it to the production environment.

Your package conversion plan may be similar to the following steps:

1. Select the packages you want to convert.
2. Migrate the packages for conversion into your test environment.
3. Prepare the packages for conversion.
4. Select test packages.
5. Analyze, investigate, and convert the test packages.
6. Test the converted applications.
7. Analyze and convert the remaining (non-test) packages.
8. Export the applications from the test environment. Import them into your production environment.

Sample plan for a limited-resource test environment

Your test environment doesn't have the resources, permissions, and architecture similar to your production environment. You can't analyze, test, and convert all of your packages. In this scenario, only analyze, investigate, convert, and test your test packages. Then migrate the remaining packages to the production environment to analyze and convert.

Your package conversion plan may be similar to the following steps:

1. Select the packages you want to convert.
2. Select test packages.
3. Migrate the test packages into your test environment.
4. Prepare the test packages for conversion.
5. Analyze, investigate, and convert the test packages.
6. Test the converted applications.
7. Export the test applications from the test environment. Then import them into your production environment.
8. Migrate the remaining packages into the production environment and prepare them for conversion.
9. Analyze, investigate, and convert the remaining packages in the production environment.
10. Release the remaining applications to the production environment.

Select and prepare packages for conversion

Select the packages that you want to convert

Not all packages are suitable to be converted into applications. Before you begin to convert packages, identify the packages that won't be converted.

The best types of package for conversion to applications are those that contain user-facing software, for example:

- Windows Installer files (.msi and .msu)
- Microsoft Application Virtualization (App-V) programs
- Windows executable files (.exe)

The types of package that are best kept as packages and not converted to applications include:

- System maintenance tools. For example, scripts or backup utilities.
- Packages for software that are out of support.

Tip

After identifying packages that aren't appropriate for conversion into applications, move them to a separate folder in the Configuration Manager console. To create a package folder in the Configuration Manager console:

- Right-click the **Packages** node.
- Select **Folders**, and then select **Create Folder**.
- Enter the folder name, for example `Not Converted`.
- Click **OK**.

Prepare the packages for conversion

For each package you want to convert, ensure that they conform to the following conditions:

- The source files location is a full UNC path, for example `\Server\Share\File`.
- Windows Installer files use only one unique product code.

Select test packages

If possible, your group of test packages should include packages that meet the following criteria:

- At least one test package with a readiness state of **Automatic**.
- At least one test package with a readiness state of **Manual**.

Ideally, your test packages should be core packages, for example:

- Packages that you know well.
- Packages that are the most important to your organization.
- Packages that you can most easily test.

Identify the packages that are appropriate for testing. Then move them to a separate folder in the Configuration Manager console.

Analyze, investigate, and convert packages

Analyze packages

To analyze an individual package or a small group, use Package Conversion Manager integrated in the Configuration Manager console. For more information, see [How to analyze and convert packages](#).

 Note

See the **Package Conversion Status** node in the **Monitoring** workspace. It displays summary information about the analysis and conversion processes.

Investigate analysis results

After analyzing the test packages, investigate the packages with a readiness state of **Manual** or **Error**. Determine the reasons why they have that state. Some common reasons for a readiness state of **Manual** or **Error** include:

- The package doesn't contain the information required to create a detection method in an application deployment type.
- The package doesn't contain the information required to convert collections to global conditions and requirements.
- The package contains more than one program.
- The package is dependent on another package that you haven't converted to an application.

For more information, use the following resources:

- Review the error messages and fixes in [Technical reference for Package Conversion Manager error messages](#)
- Review the log file **PCMTrace.log**
- [Troubleshoot Package Conversion Manager](#)

Convert the packages

For more information about how to convert packages, see [How to analyze and convert packages](#).

Note

See the **Package Conversion Status** node in the **Monitoring** workspace. It displays summary information about the analysis and conversion processes.

Test and deploy the applications

Test the applications, either in your test environment or your production environment, according to your detailed package conversion plan.

Recommendations

- Use the **Package Conversion Status** node in the **Monitoring** workspace. It displays summary information about the analysis and conversion processes.
- Investigate the programs in your packages known as wrappers. Use the Package Conversion Manager plug-in to convert their functions into the equivalent Configuration Manager functionality.
- Ensure that you thoroughly test each converted application before you deploy it in a production environment.

Next steps

[How to analyze and convert packages](#)

How to analyze and convert packages with Package Conversion Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Before you can convert a package, first analyze it. Depending on the results of the analysis, you can then do one of the following tasks:

- **Convert** the package to an application. On the **Package** list in the console, the readiness state displays **Automatic**.
- **Fix and Convert** the package, attach collections, and create global conditions. On the **Package** list in the console, the readiness state displays **Automatic**.
- **Fix and Convert** the package. On the **Package** list in the console, the readiness state displays **Manual**.
- Leave the package unconverted. On the **Package** list in the console, the readiness state displays **Not Applicable**.

How to analyze packages

1. In the Configuration Manager console, go to the **Software Library** workspace. Expand **Application Management**, and select the **Packages** node.
2. Select the package to convert. On the **Home** tab of the ribbon, in the **Package Conversion** group, select **Analyze Package**. Package Conversion Manager analyzes the package.
3. To see the readiness state of the package, add the **Readiness** column to the list of packages. The readiness state of the package determines your next action:
 - **Automatic:** [How to convert packages](#)
To also attach collections and create global conditions with an **Automatic** readiness state, see [How to fix and convert packages](#).
 - **Manual:** [How to fix and convert packages](#)
 - **Not Applicable:** This package is missing required content or a program. Add any missing content or programs and retry analysis. Or leave it in an unconverted state and continue to deploy it as a package.

- **Unknown:** First run the **Analyze** task, or wait for the next scheduled analysis. If the state doesn't change, then see [Troubleshoot Package Conversion Manager](#).

💡 Tip

Optionally, you can use the following PowerShell cmdlet to analyze a package: **Invoke-CMAnalyzePackage**.

How to convert packages

1. In the Configuration Manager console, go to the **Software Library** workspace. Expand **Application Management**, and select the **Packages** node.
2. Select the package to convert with a readiness state of **Automatic**. On the **Home** tab of the ribbon, in the **Package Conversion** group, select **Convert Package**. The Convert Package to Application wizard opens.
3. In the Convert Package to Application wizard, review the list of selected packages. Remove any packages that you don't want to convert, and select **OK**. Package Conversion Manager converts the package. The Conversion Complete window lists the Conversion Status of the new applications.

❗ Note

When you convert a package, the site records the date and time of the conversion as the UTC time.

4. Follow the instructions in the window. Select either **View applications** or **Close**.

💡 Tip

Optionally, you can use the following PowerShell cmdlet to convert a package: **Invoke-CMConvertPackage**.

How to fix and convert packages

1. In the Configuration Manager console, go to the **Software Library** workspace. Expand **Application Management**, and select the **Packages** node.

2. Select a package with a readiness state of **Manual** or **Automatic**. On the **Home** tab of the ribbon, in the **Package Conversion** group, select **Fix and Convert**.
3. In the Package Conversion Wizard, review the information on the **Package Selection** page, noting the **Items to Fix**. Then select **Next**.
4. On the **Dependency Review** page, review if the package is dependent on other listed packages, and then select **Next**.

 **Note**

If you haven't converted any of the listed dependent packages, first convert those packages. Then restart the package conversion process.

If a dependency isn't required, delete it, or ignore it and continue the conversion process.

5. On the **Deployment Type** page, review the deployment types for the new application. Change their priorities, or delete the deployment types.
6. If any of the new deployment types don't have an associated detection method, the **Detection Method** column displays a warning icon. Complete the following actions:
 - a. Select **Edit Detection Method**.
 - b. Select **Add**.
 - c. In the Detection Rule dialog box, specify a **Setting Type**.
 - d. For the specified setting type, enter the additional information required for the detection rule.
 - e. Select **OK**. If necessary, repeat this process to add multiple detection methods to each deployment type.
 - f. Select **OK**. Verify the **Detection Method** column displays an icon to confirm a correctly specified detection method.
7. Select **Next**.
8. On the **Requirements Selection** page, review the deployment types of the new application. Select a deployment type, and review the requirements for that deployment type.

Note

The wizard only displays the requirements that Package Conversion Manager converts. It doesn't convert all WQL queries in device collections to requirements.

9. Add requirements for a selected deployment type, if necessary.

10. Select **Next**.

11. Complete the wizard to create the application.

Note

When you convert a package, the site records the date and time of the conversion as the UTC time.

Monitor

Go to the **Monitoring** workspace of the Configuration Manager console, and select **Package Conversion Status**. This dashboard shows the overall analysis and conversion state of packages in the site. A new background task automatically summarizes the analysis data.

Tip

Package Conversion Manager integrated with Configuration Manager doesn't require you to schedule analysis of packages. This action is handled by the integrated summarization task. Scheduled package analysis runs every seven days by default.

Package Conversion Status

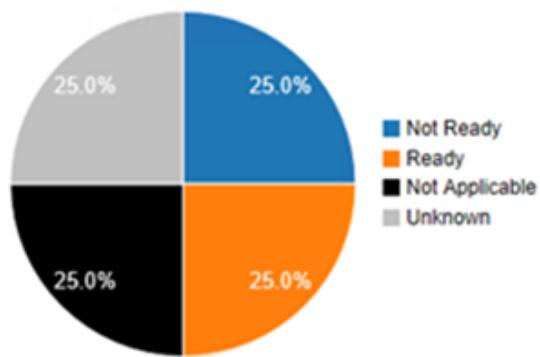
Packages Analyzed



Conversion Success



Conversion Readiness



Technical Reference for Application Deployment in Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

In this article, you'll learn how application deployments work.

Before You Begin

When troubleshooting application deployments, there are multiple items that can be useful when reviewing client logs. These items include:

- Application CI ID
- Application Unique ID
- Deployment Type Unique ID
- Application Deployment Unique ID (also known as Assignment Unique ID)
- Application Deployment Purpose
- Content Unique ID
- Collection ID and Name
- Collection Type

To simplify troubleshooting, you can run a SQL query similar to below against the Configuration Manager database to obtain the information listed above.

SQL

```
SELECT APP.CI_ID [App CI ID], APP.CI_UniqueId [App Unique ID],
APP.DisplayName [App Name],
DT.CI_UniqueId [DT Unique ID], DT.ContentId [DT Content ID],
CIA.Assignment_UniqueId [Assignment ID], CIA.CollectionID,
CIA.CollectionName,
CASE CIA.OfferTypeID WHEN 0 THEN 'Required' WHEN 2 THEN 'Available' WHEN 3
THEN 'Simulate' ELSE 'Unknown' END AS [Deployment Purpose],
CASE C.CollectionType WHEN 1 THEN 'User Collection' WHEN 2 THEN 'Device
Collection' ELSE 'Unknown' END AS [Collection Type],
DT.Technology, DT.DisplayName [DT Name]
FROM fn_ListApplicationCIs(1033) APP
JOIN fn_ListDeploymentTypeCIs(1033) DT ON DT.AppModelName = APP.ModelName
AND DT.IsLatest = 1
LEFT JOIN v_CIAssignmentToCI CIACI ON CIACI.CI_ID = APP.CI_ID
LEFT JOIN v_CIAssignment CIA ON CIACI.AssignmentID = CIA.AssignmentID
LEFT JOIN v_Collection C ON C.CollectionID = CIA.CollectionID
WHERE APP.IsLatest = 1 AND APP.DisplayName = 'Application Name' -- Replace
Application Name
```

 **Important**

When you execute this query, you **must** use the Application Name listed in the General Information tab of Application Properties, instead of using the Localized application name listed in the Software Center tab of Application properties.

Next Steps

- [Application Deployment Policy](#)

Application Deployment Policy

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Policy Creation

When you deploy an application, an instance of [SMS_ApplicationAssignment](#) class is created which represents the assignment of an application to a collection. This activity can be tracked in the [SMSProv.log](#).

text

```
SMS Provider      PutInstanceAsync SMS_ApplicationAssignment~  
SMS Provider      Auditing: User CONTOSO\Admin created an instance of class  
SMS_ApplicationAssignment.~
```

In the Configuration Manager database, this information is stored in the [CI_CIAssignments](#) table where [AssignmentType](#) 2 represents an application deployment. When the assignment is created, SMS Database Monitor component detects a change in the table then notifies Object Replication Manager to process the CI Assignment (CIA) policy. Object Replication Manager component then creates the policy for the application assignment in the database, which is stored in the [Policy](#) table in the database, and the Policy ID is based on the Application Unique ID. This activity can be tracked in the [objreplmgr.log](#) by referencing the Assignment Unique ID, which can be obtained from the SQL query referenced in the [Before You Begin](#) section.

text

```
***** Processing Application Assignment {3AC57DFE-3F87-4C59-930B-  
B9F57CB41B91} *****
```

The policy for the application assignment can be seen in the database using a SQL query similar to below.

SQL

```
SELECT P.PolicyID, PA.PolicyAssignmentID, PA.PADBID, PA.IsTombstoned,  
PA.LastUpdateTime FROM Policy P  
JOIN PolicyAssignment PA ON P.PolicyID = PA.PolicyID  
WHERE P.PolicyID = '{3AC57DFE-3F87-4C59-930B-B9F57CB41B91}' -- Replace  
Assignment Unique ID
```

Policy Targeting

After the policy is generated, the Policy Provider component assigns this policy to the resources in the collection that's targeted by the application deployment. The policy targeting information is stored in the `ResPolicyMap` table in the database. You can use the PADBID returned by the above query to track this activity in `policypv.log`. However, the PADBID recorded in the log may not always match the PADBID returned by the above query if multiple policies are getting processed simultaneously.

text

~Policy or Policy Target Change Event triggered.

~Completed batch with beginning **PADBID = 16778403** ending **PADBID = 16778403**.

ⓘ Note

`ResPolicyMap` table does not contain any targeting information for applications that are deployed as **Available** to User collections. Software Center queries a list of these applications from the Management Point, and policy targeting information for these applications is generated dynamically when a user requests an application from Software Center.

Next Steps

- [Application Deployment to Device Collections](#)
- [Application Deployment to User Collections](#)

Application Deployment for Device Collections

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

When an application is deployed to a Device collection, the policy is targeted to all the devices in the collection regardless of the deployment purpose. This article explains the policy download and deployment processing on the client.

Tip

All the information necessary to review the client logs can be obtained by running the SQL query referenced in the **Before you begin** section.

Policy Download

After the policy for the application deployment is targeted to the client, the client would download the policy at the next policy polling cycle. When the client downloads the policy, it downloads related policies in addition to the deployment policy. These related policies include the policy for the application, deployment type, global conditions, etc. Policy download activity can be tracked in the **PolicyAgent.log** on the client, using either the Application or Assignment Unique ID.

text

```
Download of policy CCM_Policy_Policy5.PolicyID="{3AC57DFE-3F87-4C59-930B-B9F57CB41B91}",PolicySource="SMS:PS1",PolicyVersion="1.00" completed (DTS Job ID: {AE88E639-0E59-40D7-AAA9-4403AAE6EE82})  
Policy state for [CCM_Policy_Policy5.PolicyID="{3AC57DFE-3F87-4C59-930B-B9F57CB41B91}",PolicySource="SMS:PS1",PolicyVersion="1.00"] is currently [Active]
```

After the policies are downloaded on the client, the Scheduler component creates schedules for deployment activation and enforcement.

Deployment Activation

Application evaluation is initiated when the deployment is activated. Scheduler component creates a schedule to activate the assignment at the Available Time

configured in the deployment. This activity can be tracked in **Scheduler.log** on the client using the Application Assignment Unique ID.

- For **Required** deployments, the activation schedule is created, but has a delay of up to two hours to avoid resource contention on Site Servers and Distribution Points. The delay helps avoid contention since application content may be downloaded during evaluation if the application is applicable based on defined Requirement Rules.

text

```
SMSTrigger '15AF8C4000080000' for scheduler 'Machine/{5F2FA409-C9B2-4100-8BC8-051820311DE1}' will fire at 08/15/2019 01:44:00 PM with randomization.
```

- For **Available** deployments, the activation schedule is created to be fired off at the Available Time configured in the Deployment.

text

```
SMSTrigger '1E4F8C4000080001' for scheduler 'Machine/{3AC57DFE-3F87-4C59-930B-B9F57CB41B91}' will fire at 08/15/2019 01:13:33 PM without randomization.
```

When the schedule time arrives, Scheduler component sends the activation message to DCM Agent to perform application evaluation.

text

```
Sending message for schedule 'Machine/{3AC57DFE-3F87-4C59-930B-B9F57CB41B91}' (Target: 'direct:DCMAgent', Name: '')
```

DCM Agent receives the activation message, and creates a job to evaluate the application.

text

```
CDCMAgent::HandleMessage - Message received for machine: '<?xml version='1.0' ?><CIAssignmentMessage MessageType='Activation'><AssignmentID>{3AC57DFE-3F87-4C59-930B-B9F57CB41B91}</AssignmentID></CIAssignmentMessage>'
```

Deployment Enforcement

Application installation is initiated when the deployment is enforced.

- For **Required** deployments, Scheduler creates a deadline schedule after policy is downloaded to enforce the application at deployment deadline. The deadline schedule isn't randomized by default. Randomization behavior for activation can be controlled by the [Disable deadline randomization](#) client setting.

text

```
SMSTrigger '15EF8C4000080000' for scheduler 'Machine/DEADLINE:{5F2FA409-C9B2-4100-8BC8-051820311DE1}' will fire at 08/15/2019  
03:05:00 PM without randomization.
```

At the deadline, Scheduler component sends the deadline message to DCM Agent.

text

```
Sending message for schedule 'Machine/DEADLINE:{5F2FA409-C9B2-4100-8BC8-051820311DE1}' (Target: 'direct:DCMAgent', Name: '')
```

DCM Agent receives the deadline message, and creates a job to enforce the application.

text

```
CDCMAgent::HandleMessage - Message received for machine: '<?xml  
version='1.0' ?><CIAssignmentMessage MessageType='EnforcementDeadline'>  
<AssignmentID>{5F2FA409-C9B2-4100-8BC8-051820311DE1}</AssignmentID>  
</CIAssignmentMessage>'
```

ⓘ Note

For deployments with deadline in the past, the application is activated and enforced immediately by the same DCM Agent job which performs the evaluation, download and installation actions.

- For **Available** deployments, there's no deadline schedule since the enforcement occurs when the application installation is initiated by the user from Software Center. When the user starts an installation, a DCM Agent job is created to perform application evaluation, download, and installation. This activity can be tracked in **DCMAgent.log** on the client.

Next Steps

- Understanding application deployment client components

Application Deployment Policy for Users

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

When an application is deployed to a User collection, the policy for the deployment is created for Required deployments only. For Available deployments, the policy is created when the user attempts to install the application from the Software Center. This article will explain the deployment process for Required as well as Available deployments.

Tip

All the information necessary to review the client logs can be obtained by running the SQL query referenced in the **Before you begin** section.

Required Deployments

The policy for a required application deployment to a User collection is targeted to all the users in the collection when the deployment is created. Client-side processing for these deployments is similar to a required deployment to a Device collection.

Deployment activation occurs at the defined Available Time, and enforcement occurs at the defined Deadline time. For more information, see [Application Deployment to Device Collections](#).

Available Deployments

Applications that are deployed to a user collection as Available behave differently. This behavior change allows the Administrator to make applications available to the users without causing resource contention for policy. When a user launches the Software Center, a list of applications that are available for the user is queried from the Management Point in real time. This request is made to the `CMUserService_WindowsAuth` virtual directory on the Management Point and can be seen in the `SCClient_[UserName].log` on the client.

text

Using endpoint Url: `https://MP.CONTOSO.COM:443/CMUserService_WindowsAuth`,
Windows authentication

When the Management Point receives this request, it queries the list of applications available to the user by executing `usp_GetApplicationPropertyValueFiltered` stored procedure. This activity can be tracked in the **UserService.log** on the Management Point.

text

```
GetFilteredApplications, startItem = 0, max rows = 60, search text = '',
filter = '', user = CONTOSO\UserName, api = 4.0, source =
UserService_WinAuth_SoftwareCenter, platform = <OSPlatform>
GetFilteredApplications: returned 1 rows out of 1 total
```

Software Center receives the list and displays the applications that the user can install. When the user clicks on the application, additional information about the application is queried from the Management Point, which involves execution of stored procedures such as `usp_GetApplicationInfo`, `usp_GetAppModelApplicationSupersedence`, `usp_GetDeploymentTypeForAnApp`, etc.

The deployment is activated when the user selects the application and clicks on the **Install** button, and a DCM Agent Job is created to evaluate the application. If the application is applicable, another DCM Agent Job is created to download and enforce the application. This activity can be tracked in the **DCMAgent.log** on the client.

Next Steps

- [Understanding application deployment client components](#)

Understanding Application Deployment Client Components

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Application deployment evaluation and enforcement operations are handled by the DCM Agent and CI Agent components on the client. This article explains how a typical DCM and CI Agent job operates.

DCM Agent

DCM Agent is the high-level client component responsible for evaluation of configuration items, which includes applications. When a deployment is activated or enforced, a DCM Agent job is created which reads the assignment policy and determines the actions that need to be performed. This activity can be tracked in the **DCMAgent.log** on the client using the DCM Agent Job ID, which can be identified by looking for the Application Unique ID.

Device Deployments

- For **Required** deployments, DCMAgent.log would show the applicable actions. These actions may differ depending on whether the deployment deadline has already passed.

text

```
# Evaluation Job example:  
DCMAgentJob({A9E850E2-91B0-4122-94FD-D14EDF925AF7}):  
CDCMAgentJob::PopulateCIsFromAssignment - CI policy  
Id:ScopeId_B63CEBE7-8A69-4FBE-994F-  
5AD0A8488D27/RequiredApplication_fc76ef0a-3ab0-4110-8cce-1addc36d0225  
version:3 with actions: Evaluation, Content Download  
  
# Enforcement Job example:  
DCMAgentJob({4C8A9F6E-390B-450E-B505-B5698DB68EDD}):  
CDCMAgentJob::PopulateCIsFromAssignment - CI policy  
Id:ScopeId_B63CEBE7-8A69-4FBE-994F-  
5AD0A8488D27/RequiredApplication_fc76ef0a-3ab0-4110-8cce-1addc36d0225  
version:3 with actions: Evaluation, Install, Uninstall, Update, Look-  
ahead Install, Look-ahead Uninstall, Look-ahead Update
```

- For **Available** deployments, DCMAgent.log shows that the deployment **is not mandatory**. For these deployments, application evaluation is done but enforcement is skipped unless the user initiated the installation.

text

```
# Evaluation Job example:
DCMAgentJob({E353BF94-D7ED-4ADD-AF0F-9273F6A67FC1}):
CDCMAgentJob::PopulateCIsFromAssignment - [SCAN] CI policy Id
:ScopeId_B63CEBE7-8A69-4FBE-994F-
5AD0A8488D27/RequiredApplication_fc76ef0a-3ab0-4110-8cce-1addc36d0225
version:3 - Assignment:{3AC57DFE-3F87-4C59-930B-B9F57CB41B91} is not
mandatory.

# Enforcement Job (user initiated) example:
Request to enforce application ConfigMgr Toolkit(ScopeId_B63CEBE7-8A69-
4FBE-994F-5AD0A8488D27/Application_fc76ef0a-3ab0-4110-8cce-
1addc36d0225.3) immediately for target: machine with action(s):
Evaluation, Install, Update
CDCMAgentJobMgr::CreateInteractiveJob - Queuing new job: {D331249E-
F7DE-481B-A497-8E8B5E7B91C3}
```

User Deployments

- For **Required** deployments, DCMAgent.log would show the applicable actions. These actions may differ depending on whether the deployment deadline has already passed.

text

```
# Evaluation Job example:
DCMAgentJob({65D9688D-1781-4DA3-B07A-193D481251C6}):
CDCMAgentJob::PopulateCIsFromAssignment - CI policy
Id:ScopeId_C8F7EAE6-DBA8-4970-B3FF-
47ED706868DE/RequiredApplication_6b39398b-fd20-47ca-bd68-074274509f98
version:2 with actions: Evaluation, Content Download

# Enforcement Job example:
DCMAgentJob({2B0DA272-FC65-4F31-9557-C4D840D650F1}):
CDCMAgentJob::PopulateCIsFromAssignment - CI policy
Id:ScopeId_C8F7EAE6-DBA8-4970-B3FF-
47ED706868DE/RequiredApplication_6b39398b-fd20-47ca-bd68-074274509f98
version:2 with actions: Evaluation, Install, Uninstall, Update, Look-
ahead Install, Look-ahead Uninstall, Look-ahead Update
```

- For **Available** deployments, DCM Agent jobs are created for evaluation and enforcement when the application installation is initiated by the user.

text

```
# Evaluation Job example:  
DCMAgentJob({FBB44C84-DB06-41F7-8DC1-D9BA368F0C20}):  
CDCMAgentJob::PopulateCIsFromAssignment - [SCAN] CI policy Id  
:ScopeId_C8F7EAE6-DBA8-4970-B3FF-  
47ED706868DE/RequiredApplication_6b39398b-fd20-47ca-bd68-074274509f98  
version:2 - Assignment:{7EA17128-EB4F-448A-88A7-B865E7DA228C} is not  
mandatory.  
  
# Enforcement Job example:  
CAppMgmtSDK::EnforceAppPolicy ScopeId_C8F7EAE6-DBA8-4970-B3FF-  
47ED706868DE/RequiredApplication_6b39398b-fd20-47ca-bd68-074274509f98.  
CDCMAgentJobMgr::CreateInteractiveJob - Queuing new job: {7936D7F3-  
24B0-401D-BADD-59EB5B49C2C2}
```

CI Agent

CI Agent is the client component responsible for evaluation and remediation of configuration items. DCM Agent reads the assignment policy and creates a job for the CI Agent component to perform the requested actions. **DCMAgent.log** records the CI Agent Job ID, which is useful for tracking the CI Agent activity in the **CIAgent.log** on the client.

text

```
DCMAgentJob({E353BF94-D7ED-4ADD-AF0F-9273F6A67FC1}):  
CDCMAgent::InitiateCIAgentJob - Starting CI Agent Job {57AF6FA1-3482-4469-  
9881-A63F41D18406} for target: machine. Refer to this CI agent job ID in  
ciagent.log for more details
```

A typical CI Agent job goes through multiple phases, which can be identified by filtering **CIAgent.log** on the CI Agent Job ID and then looking for **TransitionState**. Some of the key phases for an application deployment CI Agent job are:

- **DownloadingCIs**
 - During this phase, application metadata required to evaluate the application is downloaded. The metadata includes detection method, requirement rules, global conditions, etc. This activity can be tracked in **CIDownloader.log** and **DataTransferService.log**. For **Available** deployments, this process occurs during the first evaluation of the application. For **Required** deployments however, this process occurs immediately after the policy is downloaded.
- **InvokingSdmMethod**

- During this phase, the application detection method is used to check if the application is installed and the desired state is determined. This activity can be tracked in **AppDiscovery.log** and **AppIntentEval.log**. For more information about this phase, see [Application Evaluation](#).
- **StateDownloadingContents**
 - During this phase, application content is downloaded if necessary. This activity can be tracked in **CAS.log**, **ContentTransferManager.log**, **LocationServices.log**, and **DataTransferService.log**. For more information about this phase, see [Application Download](#).
- **StateEnforcingCIs**
 - During this phase, the application installation is initiated. This activity can be tracked in **AppEnforce.log**. For more information about this phase, see [Application Installation](#).
- **StateEnforcementReporting**
 - During this phase, application installation state is recorded for reporting to the Management Point. This activity can be tracked in **StateMessage.log**.

Although the CI Agent job goes through all the phases, it skips the phase if it isn't required. As an example, for **Available** deployments StateDownloadingContents and StateEnforcingCIs phases are skipped until the user attempts to install the application from Software Center. However, for **Required** deployments, the StateDownloadingContents phase downloads application content (if necessary) when the assignment is activated, but the StateEnforcingCIs phase is skipped if the deadline is in the future. This behavior can be observed in the **CIAgent.log** by filtering on the CI Agent Job ID and looking for **Skipping policy**.

text

```
{57AF6FA1-3482-4469-9881-A63F41D18406} - Skipping policy CI <CI Unique ID> and all dependents for ContentDownload task since CI action was not requested.
{57AF6FA1-3482-4469-9881-A63F41D18406} - Skipping policy CI <CI Unique ID> and all dependents for Enforce task since CI action was not requested.
```

Next Steps

- [Application Evaluation](#)
- [Application Download](#)
- [Application Installation](#)

Application Deployment Evaluation

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Before you continue, please review [Application deployment client components](#) to understand DCM and CI Agent job processing.

Application evaluation is performed by the DCM Agent and CI Agent components when the deployment is activated. To understand when the assignment is activated, see the [Application Deployment to Device Collections](#) or [Application Deployment to User Collections](#) articles.

Application Detection and Evaluation

Application evaluation is performed during the **InvokingSdmMethod** phase of a CI Agent job. This phase is where the client evaluates the detection method defined for the application to determine if the application is installed on the device. This activity can be tracked in **AppDiscovery.log** using the Deployment Type Unique ID or Deployment Type Name.

text

```
Performing detection of app deployment type ConfigMgr Toolkit - Windows
Installer (*.msi file)(ScopeId_B63CEBE7-8A69-4FBE-994F-
5AD0A8488D27/DeploymentType_1d49ef88-cf3b-42fa-b198-388d220ccb44, revision
2) for system.
+++ Did not detect app deployment type ConfigMgr Toolkit - Windows Installer
(*.msi file)(ScopeId_B63CEBE7-8A69-4FBE-994F-
5AD0A8488D27/DeploymentType_1d49ef88-cf3b-42fa-b198-388d220ccb44, revision
2) for system.
```

ⓘ Note

Above example shows detection for an MSI application where the detection is done by checking if the MSI Product Code is installed on the device. For applications using alternate detection methods, the appropriate detection method is used to check if the application is installed.

Next, the client evaluates the desired state of the application based on the Deployment Purpose. This step also involves detecting whether the application has any dependencies

or supersedence rules that should be honored for the application. This activity can be tracked in **ApplIntentEval.log** using the Application and Deployment Type Unique ID.

text

```
# Available Application Deployment

[Application or DT Unique ID] :- Current State = NotInstalled, Applicability = Applicable, ResolvedState = Available, ConfigureState = NotNeeded, Title = [Application or DT Name]

# Required Application Deployment

[Application or DT Unique ID] :- Current State = NotInstalled, Applicability = Applicable, ResolvedState = Installed, ConfigureState = NotNeeded, Title = [Application or DT Name]

# Requirement Rules Not Met

[Application or DT Unique ID] :- Current State = NotInstalled, Applicability = NotApplicable, ResolvedState = None, ConfigureState = NotNeeded, Title = [Application or DT Name]
```

In the log entry above, **Current State** indicates whether the application is currently installed on the device. **Applicability** indicates whether the application is applicable based on defined requirement rules. **ResolvedState** indicates the desired state of the application based on the deployment purpose.

💡 Tip

Use the **Deployment Monitoring Tool** to view the application state, applicability state and requirement violations.

Next Steps

- [Application Download](#)

Application download in Configuration Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Before you continue, review [Application deployment client components](#) to understand DCM and CI Agent job processing.

Download initiation

Application content download is started by the CI Agent component on the client during the **StateDownloadingContents** phase. This process is the same, regardless of whether the application is deployed to a Device Collection or a User collection.

- For **Available** deployments, application content is downloaded when the user starts the application installation from Software Center.
- For **Required** deployments, application content is downloaded when the assignment is activated and the application is found Applicable after evaluation. To understand when the assignment is activated, see the [Application Deployment to Device Collections](#) or [Application Deployment to User Collections](#) articles.

When CI Agent starts the content download, it creates a task that is handled by the CI Task Manager component. CI Task Manager then starts the content download. This activity can be tracked in the **CITaskMgr.log** by using the Deployment Type Unique ID.

text

```
Initiating task ContentDownload for CI ScopeId_B63CEBE7-8A69-4FBE-994F-5AD0A8488D27/DeploymentType_1d49ef88-cf3b-42fa-b198-388d220ccb44.2  
(ConfigMgr Toolkit - Windows Installer (*.msi file)) for target: , consumer:  
{53EA65C2-D596-4215-83E4-F7007B78E18C}
```

Distribution Point Location

All download tasks are handled by Content Access component, which is responsible for managing the client cache. After the download task is created, Content Access component checks if the content is already available in the client cache. If the content isn't available, it creates a location request to get a list of Distribution Points from where

the content can be obtained. This activity can be tracked in **CAS.log** and **LocationServices.log** on the client using the Content Unique ID.

text

```
Requesting locations synchronously for content Content_00a8f9e6-8e44-42f5-a0ef-9b5c86a88498.1 with priority Foreground  
ContentLocationRequest : <Request XML Body>  
Reply Message Body : <Reply XML Body>
```

Important

Although Location Services component handles the location requests, it doesn't directly request locations from the Management Point. All requests to the Management Point typically go through CCM Messaging component, which logs to **CcmMessaging.log**.

Location reply XML contains the list of distribution points based on the client's boundary group. This list is parsed and persisted in WMI on the client according to the **Content Source Priority**. This activity can be seen in **ContentTransferManager.log**, by using the Content Unique ID and looking for **Persisted location**.

If the location reply XML doesn't contain any distribution points, **ContentTransferManager.log** would show **Received empty location update** and the client may get stuck at 0% while downloading the application. This reply can typically occur because of boundary group configuration issues. For more information, see [Download failures](#).

Content Download

Once the Distribution Point locations are obtained, Content Access component creates a Content Transfer job. This activity can be tracked in **CAS.log** using the Content Unique ID.

text

```
Submitted CTM job {6D0EA720-EB4E-4893-8395-8B27470A6CFB} to download Content  
Content_00a8f9e6-8e44-42f5-a0ef-9b5c86a88498.1 under context System
```

Content Transfer Manager then creates a Data Transfer Service job to do the content download. This activity can be tracked in **ContentTransferManager.log** on the client using the Content Unique ID.

text

```
CTM job {6D0EA720-EB4E-4893-8395-8B27470A6CFB} (corresponding DTS job {708C7F21-8898-49AB-900E-BA6E5F1A39BC}) started download from '<Distribution Point URL>/Content_00a8f9e6-8e44-42f5-a0ef-9b5c86a88498.1' for full content download.
```

ⓘ Note

This log entry can be used to identify the CTM and DTS job ID's, which can be used to track the progress of the Content Transfer in **ContentTransferManager.log** and **DataTransferService.log** respectively.

Data Transfer Service downloads the application content by creating a Background Intelligent Transfer Service (BITS) job and waiting for the download to complete. This activity can be tracked in **DataTransferService.log** on the client using the DTS Job ID obtained from **ContentTransferService.log**.

text

```
Starting BITS job '{40263E01-2EDD-462F-ABBA-A5E892CB9229}' for DTS job '{708C7F21-8898-49AB-900E-BA6E5F1A39BC}' under user 'S-1-5-18'.  
DTSJob {708C7F21-8898-49AB-900E-BA6E5F1A39BC} in state 'DownloadingData'.  
DTS job {708C7F21-8898-49AB-900E-BA6E5F1A39BC} has completed
```

After the download is complete, Content Access component is notified. Content Access component then verifies the downloaded content to ensure that the content wasn't altered during download. This activity can be tracked in **CAS.log** using the Content Unique ID.

text

```
Hash verification succeeded for content Content_00a8f9e6-8e44-42f5-a0ef-9b5c86a88498.1 downloaded under context System
```

Finally, after content is verified, CI Agent receives the task complete notification and the CI Agent job moves to the next phase.

text

```
CIAgentJob({2BF84225-C9E8-49A6-A308-A160C4B799D3}):  
CAgentJob::HandleEvent(Event=CTaskComplete,  
CurrentState=StateDownloadingContents)
```

Next steps

[Application Installation](#)

Application Installation

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Before you continue, please review [Application deployment client components](#) to understand DCM and CI Agent job processing.

Application installation is performed by DCM Agent and CI Agent components when the deployment is enforced. The enforcement time differs for Available and Required deployments. To understand when the assignment is enforced, see the [Application Deployment to Device Collections](#) or [Application Deployment to User Collections](#) articles.

Enforcement Initiation

Application installation is initiated by the CI Agent component on the client during the **StateEnforcingCIs** phase. This process is the same, regardless of whether the application is deployed to a Device Collection or a User collection.

- For **Available** deployments, the application is installed when the user initiates the application installation from Software Center.
- For **Required** deployments, the application is installed at deployment deadline. However, the user can initiate the installation from Software Center before the deadline.

When CI Agent initiates the application installation, it creates a task that is handled by the CI Task Manager component. CI Task Manager then initiates the installation. This activity can be tracked in the **CITaskMgr.log** by using the Deployment Type Unique ID.

text

```
Initiating task Enforce for CI ScopeId_B63CEBE7-8A69-4FBE-994F-  
5AD0A8488D27/DeploymentType_1d49ef88-cf3b-42fa-b198-388d220ccb44.2  
(ConfigMgr Toolkit - Windows Installer (*.msi file)) for target: , consumer:  
{9BC3154A-98F1-4595-A967-173D536A3F94}  
Initiated application enforcement. : CITask(ScopeId_B63CEBE7-8A69-4FBE-994F-  
5AD0A8488D27/DeploymentType_1d49ef88-cf3b-42fa-b198-  
388d220ccb44.2..Install.Enforce)
```

Application Enforcement

After the application enforcement is initiated, the client performs the application detection again to ensure the application isn't already installed. Once it's determined that the application isn't installed, the application installation is initiated. This activity can be tracked in the **AppEnforce.log** on the client using the Deployment Type Unique ID.

text

```
+++ Starting Install enforcement for App DT "ConfigMgr Toolkit - Windows
Installer (*.msi file)" ApplicationDeliveryType - ScopeId_B63CEBE7-8A69-
4FBE-994F-5AD0A8488D27/DeploymentType_1d49ef88-cf3b-42fa-b198-388d220ccb44,
Revision - 2, ContentPath - C:\WINDOWS\ccmcache\2, Execution Context -
System
    Executing Command line: "C:\WINDOWS\system32\msiexec.exe" /i
"ConfigMgrTools.msi" /q /qn with user context
    Process 7292 terminated with exitcode: 0
Status is switching to Success
```

Installation Verification

After the application is installed, the application detection method is used again to ensure that the application was detected as installed.

text

```
Performing detection of app deployment type ConfigMgr Toolkit - Windows
Installer (*.msi file)(ScopeId_B63CEBE7-8A69-4FBE-994F-
5AD0A8488D27/DeploymentType_1d49ef88-cf3b-42fa-b198-388d220ccb44, revision
2) for system.
+++ Discovered MSI application [AppDT Id: ScopeId_B63CEBE7-8A69-4FBE-994F-
5AD0A8488D27/DeploymentType_1d49ef88-cf3b-42fa-b198-388d220ccb44, Revision:
2, MSI Product code: {4FFF7ECC-CCF7-4530-B938-E7812BB91186}, MSI Product
version: ]
++++++ App enforcement completed (3 seconds) for App DT "ConfigMgr Toolkit -
Windows Installer (*.msi file)" [ScopeId_B63CEBE7-8A69-4FBE-994F-
5AD0A8488D27/DeploymentType_1d49ef88-cf3b-42fa-b198-388d220ccb44], Revision:
2, User SID: ] ++++++
```

Finally, after enforcement is complete, CI Agent receives the task complete notification and the CI Agent job moves to the next phase.

text

```
CIAgentJob({2BF84225-C9E8-49A6-A308-A160C4B799D3}):
CAgentJob::HandleEvent(Event=CTaskComplete, CurrentState=StateEnforcingCIs)
```

Next Steps

- Troubleshoot application deployments
- Common error codes for app installation

Application installation common error codes reference

Article • 02/22/2023

Applications can be installed on clients by creating deployments from the Configuration Manager console or by [targeting applications to tenant attached devices](#) from the [Microsoft Intune admin center](#). Use the information in this article to assist with troubleshooting application installation errors.

General troubleshooting tips

Generally, if an application installs successfully on a device with the given command line in the system context, it will install successfully through Configuration Manager and from the Microsoft Intune admin center. You can simulate this by using [PSEXEC](#).

1. Open an administrative command prompt.
2. Change directory to where you saved [PSEXEC](#).
3. Type in `psexec -accepteula -s -i cmd`.
4. This opens a new command prompt window running interactively in the system context. Check that you're in the system context by running a `whoami` command.
5. Run the install from the new windows with the installation command line. For example, `msiexec /i "My App.msi" /q` would be a quiet install of the "My App" msi file.

You may also find that searching through multiple files for a specific string is useful. For instance, you might want to search all the client `.mof` files for a specific class, or you might want to search logs for a specific ID. Using a specific ID when searching can give you an understanding of how components are related to each other. Use the [select-string cmdlet](#) in those instances.

PowerShell

```
select-string -Path "c:\windows\ccm\*.mof" -Pattern 'CacheInfoEx'  
select-string -Path "c:\windows\ccm\logs\*.log" -Pattern  
'CacheInfoEx.CacheId="ccfe8120-4b9b-4f6e-b8fb-f8c1b1fd74d8'
```

Configuration Manager errors

Error code	Error source	Error message
------------	--------------	---------------

Error code	Error source	Error message
0x87D00202	Configuration Manager	Service is shutting down
0x87D00207	Configuration Manager	Parsing error
0x87D00213	Configuration Manager	Timeout occurred
0x87D00215	Configuration Manager	Item not found
0x87D00235	Configuration Manager	Syntax error occurred while parsing
0x87D00244	Configuration Manager	The object or subsystem has not been initialized
0x87D0027C	Configuration Manager	CI documents download timed out
0x87D00289	Configuration Manager	Failed to decompress CI documents
0x87D00314	Configuration Manager	CI Version Info timed out
0x87D00321	Configuration Manager	The script execution has timed out
0x87D00324	Configuration Manager	The application was not detected after installation completed
0x87D00325	Configuration Manager	Application was still detected after uninstall completed
0x87D00327	Configuration Manager	Script is not signed
0x87D00329	Configuration Manager	Application requirement evaluation or detection failed
0x87D00607	Configuration Manager	Content not found
0x87D00667	Configuration Manager	No current or future service window exists to install software updates
0x87D01106	Configuration Manager	Failed to verify the executable file is valid or to construct the associated command line

Error code	Error source	Error message
0x87D01107	Configuration Manager	Failed to access all the provided program locations. This program may retry if the maximum retry count has not been reached
0x87D01201	Configuration Manager	The content download cannot be performed because there is not enough available space in cache or the disk is full
0x87D01202	Configuration Manager	The content download cannot be performed because the total size of the client cache is smaller than the size of the requested content
0x87D01281	Configuration Manager	A supported App-V client is not installed
0x87D0128F	Configuration Manager	The App-V sftmim command returned failure
0x87D01290	Configuration Manager	An error occurred when querying the App-V WMI provider
0x87D103E8	Configuration Manager	Error Unknown
0x87D1076C	Configuration Manager	Application was successfully installed

General Configuration Manager troubleshooting tips

When an application fails to install and the error source is **Configuration Manager**, typically, following the [application troubleshooting guide](#) and using the [general troubleshooting tips](#) helps you resolve the error. You may also want to use [Support Center for Configuration Manager](#) to help troubleshoot and review information about your clients.

0x87D00202

Message: Service is shutting down

Additional information for error resolution: Verify that the Configuration Manager client is running on the target device. Verify the client is running by:

- Reviewing the **CCMExec.log** on the device
- Verifying that the **SMS Agent Host** service is running on the device

0x87D00207

Message: Parsing error

Additional information for error resolution: This error generally occurs in one of the Configuration Manager components when a piece of data is invalid. This error could stem from something missing for the application, an old package version, or a number of other general errors. Follow the [application troubleshooting guide](#) to help locate the error and resolve it. It may be necessary to review additional logs for components that support application installation. Searching for specific IDs or error codes in the logging may help you identify the problem. For more information, see [general troubleshooting tips](#).

0x87D00213

Message: Timeout occurred

Additional information for error resolution: Increase the [Maximum allowed run time \(minutes\)](#) for the application. Ensure that the maintenance window on the client is large enough to support the runtime. For more information, see the [application troubleshooting guide](#) to help resolve the error.

0x87D00215

Message: Item not found

Additional information for error resolution:

Verify that the following exist and are accessible to the client:

- The [application deployment](#) exists and the client sees the policy.
- The [application content exists and is available to the client](#)

For more information, see the [application troubleshooting guide](#) to help resolve the error.

0x87D00235

Message: Syntax error occurred while parsing

Additional information for error resolution: This error generally occurs in one of the Configuration Manager components when a piece of data is invalid. This error could stem from something missing for the application, an old package version, or a number of other general errors. Follow the [application troubleshooting guide](#) to help locate the error and resolve it. It may be necessary to review additional logs for components that

support application installation. Searching for specific IDs or error codes in the logging may help you identify the problem. For more information, see [general troubleshooting tips](#).

0x87D00244

Message: The object or subsystem has not been initialized

Additional information for error resolution: This error generally occurs in one of the Configuration Manager components when a piece of data is invalid. This error could stem from something missing for the application, an old package version, or a number of other general errors. Follow the [application troubleshooting guide](#) to help locate the error and resolve it. It may be necessary to review additional logs for components that support application installation. Searching for specific IDs or error codes in the logging may help you identify the problem. For more information, see [general troubleshooting tips](#).

0x87D0027C

Message: CI documents download timed out

Additional information for error resolution: The CI documents activity can be tracked in `CIAgent.log`, `CIDownloader.log`, and `DataTransferService.log`. For more information, see the [CI Agent section](#) of the application troubleshooting guide.

0x87D00289

Message: Failed to decompress CI documents

Additional information for error resolution: The CI documents activity can be tracked in `CIAgent.log`, `CIDownloader.log`, and `DataTransferService.log`. For more information, see the [CI Agent section](#) of the application troubleshooting guide.

0x87D00314

Message: CI Version Info timed out

Additional information for error resolution: Typically this error occurs when a change was made to the application and the client doesn't have the new information for it. Verify that the client is [getting the policy](#) and it knows about any [updated revisions](#) to the application.

0x87D00321

Message: The script execution has timed out

Additional information for error resolution: Check the [AppEnforce.log](#) for details. You may need to increase the [Maximum allowed run time \(minutes\)](#) for the application. Ensure that the maintenance window on the client is large enough to support the run time. For more information, see the [application troubleshooting guide](#) to help resolve the error.

0x87D00324

Message: The application was not detected after installation completed

Additional information for error resolution: Review the [AppDiscovery.log](#) and the [CIAgent.log](#). Once an installation is completed, the [application detection](#) is used again to [verify the installation](#).

0x87D00325

Message: Application was still detected after uninstall completed

Additional information for error resolution: Verify the correct uninstall command was used in the [AppEnforce.log](#). Review the [AppDiscovery.log](#) and the [CIAgent.log](#). Once an uninstall is completed, the [application detection](#) is used again to [verify the uninstall](#).

0x87D00327

Message: Script is not signed

Additional information for error resolution: Verify the [PowerShell execution policy client setting](#) for the device. The default for this client setting is [AllSigned](#) so an unsigned script will cause a failure.

0x87D00329

Message: Application requirement evaluation or detection failed

Additional information for error resolution: Review the [AppIntentEval.log](#) to discover dependencies and supersedence rules for the application and their states. For more information, see [Application deployment evaluation](#).

0x87D00607

Message: Content not found

Additional information for error resolution: Verify the content for the application is on a distribution point and that the distribution point is accessible to the client. For more information, see [Application download in Configuration Manager](#).

0x87D00667

Message: No current or future service window exists to install software updates

Additional information for error resolution: Ensure that the [maintenance window](#) on the client is large enough to support the [Maximum allowed run time \(minutes\)](#) for the application installation and that the client has received the policy for the window.

0x87D01106

Message: Failed to verify the executable file is valid or to construct the associated command line

Additional information for error resolution: Verify that the executable file is installable on its own then verify it's installable with the given command line.

0x87D01107

Message: Failed to access all the provided program locations. This program may retry if the maximum retry count has not been reached

Additional information for error resolution: The client is getting locations for the content, but can't reach the locations. Review the client's [LocationServices.log](#) for the [Distribution Point=](#). Use [ContentTransferManager.log](#) and [DataTransferService.log](#) to monitor the download for errors.

0x87D01201

Message: The content download cannot be performed because there is not enough available space in cache or the disk is full

Additional information for error resolution: Check that the machine has enough space on the drive. Compare the size of the `ccmcache` directory with the [client cache settings](#) and ensure the setting is adequate for the application's size.

0x87D01202

Message: The content download cannot be performed because the total size of the client cache is smaller than the size of the requested content

Additional information for error resolution: Compare the size of the `ccmcache` directory with the [client cache settings](#) and ensure the setting is adequate for the application's size.

0x87D01281

Message: A supported App-V client is not installed

Additional information for error resolution: Verify that a [supported version](#) of App-V is installed on the client.

0x87D0128F

Message: The App-V sftmime command returned failure

Additional information for error resolution: For information on sftmime commands, see [Manage Virtual Applications by Using the Command Line](#).

0x87D01290

Message: An error occurred when querying the App-V WMI provider

Additional information for error resolution: For information on the App-V WMI provider, see [Application Virtualization Client WMI Provider](#).

0x87D103E8

Message: Error Unknown

Additional information for error resolution: Follow the [application troubleshooting guide](#) to help locate the error and resolve it. It may be necessary to review additional logs for components that support application installation. Searching for specific IDs or error codes in the logging may help you identify the problem. For more information, see [general troubleshooting tips](#).

0x87D1076C

Message: Application was successfully installed

Additional information for error resolution: The application was successfully installed.

MSI errors

Error code	Error source	Error message
1602	MSI	User cancel installation
1603	MSI	Fatal error during installation
1605	MSI	This action is only valid for products that are currently installed
1618	MSI	Another program is being installed. Please wait until that installation is complete, and then try installing this software again
1633	MSI	This installation package is not supported by this processor type. Contact your product vendor
1638	MSI	Another version of this product is already installed. Installation of this version cannot continue. To configure or remove the existing version of this product, use Add/Remove Programs on the Control Panel
1642	MSI	The upgrade patch cannot be installed by the Windows Installer service because the program to be upgraded may be missing, or the upgrade patch may update a different version of the program. Verify that the program to be upgraded exists on your computer and that you have the correct upgrade patch

General MSI troubleshooting tips

When errors are encountered from MSI, typically you'll need to [Enable Windows Installer logging](#). After the logging is enabled, you can retry the problem installation and Windows Installer will track the progress and post it to the `%temp%` folder. The new log's file name is random. However, the first letters are `Msi` and the file name has a `.log` extension.

The [MsiExec.exe and InstMsi.exe Error Messages](#) and [Windows Installer Action Return Values](#) lists are useful when reviewing a Windows Installer log as are the [general troubleshooting tips](#).

1602

Message: User cancel installation

Additional information for error resolution: The installation was canceled by the user. Ask the user to install the application fully. If possible, you can attempt to run the installation for the system rather than the user.

1603

Message: Fatal error during installation

Additional information for error resolution: [Enable Windows Installer logging](#) and run the install again. When reviewing the installer log, typically an entry stating `Return value 3` is located near the failure reason in the log. For more information on possible return values and their meaning, see [Windows Installer Action Return Values](#).

1605

Message: This action is only valid for products that are currently installed

Additional information for error resolution: Ensure that the product is installed before running a dependant install.

1618

Message: Another program is being installed. Please wait until that installation is complete, and then try installing this software again

Additional information for error resolution: Wait for the prior installation to complete before running a new one. If the prior installation stops responding, you can attempt to stop the installation or terminate the process. Terminating a process might have undesired results.

1633

Message: This installation package is not supported by this processor type. Contact your product vendor

Additional information for error resolution: Ensure that the device's processor architecture is appropriate for the software. Verify the target device meets or exceeds the minimum processor requirement for the application. Contact the product vendor if the device's processor meets the product's processor support specifications.

1638

Message: Another version of this product is already installed. Installation of this version cannot continue. To configure or remove the existing version of this product, use Add/Remove Programs on the Control Panel

Additional information for error resolution: Uninstall the the unwanted version of the product. If you aren't using Configuration Manager, a script, or another management tool to uninstall, uninstall from the device manually. For Windows 10 or later clients, use **Windows Settings > Apps** to uninstall the unwanted version of the product. For earlier versions of Windows, use **Programs and Features** from the Control Panel to uninstall the unwanted version of the product.

1642

Message: The upgrade patch cannot be installed by the Windows Installer service because the program to be upgraded may be missing, or the upgrade patch may update a different version of the program. Verify that the program to be upgraded exists on your computer and that you have the correct upgrade patch

Additional information for error resolution: Verify the device meets the product versioning prerequisites for the installation.

Windows errors

Error code	Error source	Error message
1	Windows	Incorrect function
2	Windows	The system cannot find the file specified
692	Windows	Debugger terminated process
0x80000003	Windows	One or more arguments are invalid
0x80000007L	Windows	Operation aborted
0x80000009	Windows	General access denied error
0x80004005	Windows	Unspecified error
0x8000FFFF	Windows	Catastrophic failure
0x80040154	Windows	Class not registered
0x80091007	Windows	The hash value is not correct

Error code	Error source	Error message
0xC0000142	Windows	Initialization of the dynamic link library failed. The process is terminating abnormally

General Windows troubleshooting tips

Use the [Windows system error codes](#) list or [Download the Microsoft Error Lookup Tool](#) for looking up additional codes that aren't listed in this article. Using the Windows event logs and the [general troubleshooting tips](#) can also help identify the cause of these errors.

1

Message: Incorrect function

Additional information for error resolution: Review the Windows event logs around the time of the failure in combination with the installation logs to determine the possible cause of the error.

2

Message: The system cannot find the file specified

Additional information for error resolution:

- If the missing file is a system file, run the [System File Checker tool to repair missing or corrupted system files](#). You can also use `/scanfile=file` or `/verifyfile` with the [sfc command](#) to scan the binary and check if there is any issue with that file.
- If the missing file is an application file, you can repair or uninstall and reinstall the application to replace the missing file.
- If you're unsure which file is missing and the logs aren't listing it, you may want to use [Process Monitor](#) to help identify the problematic file.
 - You can launch Process Monitor without capturing events and filters by using `ProcMon.exe /NoConnect /NoFilter /AcceptEULA`

692

Message: Debugger terminated process

Additional information for error resolution: Detach any debuggers attached to the process and retry the application installation.

0x80000003

Message: One or more arguments are invalid

Additional information for error resolution: Review the Windows event logs around the time of the failure in combination with the installation logs to determine the possible cause of the error.

0x80000007L

Message: Operation aborted

Additional information for error resolution: Use the installation logs and Configuration Manager application logs to determine why installation stopped. Merge the logs so you can easily review what happened before the 0x80000007L error. Use `eventvwr.msc` to review the Windows event logs for additional events that occurred around the time of the installation failure.

0x80000009

Message: General access denied error

Additional information for error resolution: If the issue isn't clear from the logs, using `eventvwr.msc` to review Windows event logs and [Process Monitor](#) can help identify problematic files or processes. If needed, use the Windows user interface or `icacls` to modify permissions on the problematic file.

Additional tips for file permissions in Windows operating systems:

- Deny permissions always take precedence over Allow permissions.
- Explicit permissions take precedence over inherited permissions.
- If NTFS permissions conflict, or example, if group and user permissions are contradictory, the most liberal permissions take precedence.
- Permissions are cumulative.

0x80004005

Message: Unspecified error

Additional information for error resolution: Use the installation logs and Configuration Manager application logs to determine why installation stopped. Merge the logs so you can easily review what happened before the 0x80004005 error. Use `eventvwr.msc` to review the Windows event logs for additional events that occurred around the time of the installation failure. Follow the [application troubleshooting guide](#) to help resolve the error. [Process Monitor](#) can also help identify the failure.

0x8000FFFF

Message: Catastrophic failure

Additional information for error resolution: Review the Windows event logs around the time of the failure in combination with the installation logs to determine the possible cause of the error.

0x80040154

Message: Class not registered

Additional information for error resolution: This is typically a configuration-related DCOM error. Review DCOM configuration settings using `dcomconfig`. If there's a problematic .dll file, you can use `regsvr32` to register the dll file and try the install again. A large number of problematic files could be a sign of an underlying issue that needs to be resolved before you can install the application.

0x80091007

Message: The hash value is not correct

Additional information for error resolution: The hash of a file isn't correct and the installation can't complete. Typically you will see this error in the `CAS.log`. Check to see if file contents for the application were recently updated. There may be an issue with the package, in some cases you may need to rebuild and redistribute it. This issue can also happen if there is a sharing violation on a file, such as a security application scanning the file. Configuration Manager expects exclusive access to the file during a hash check. You can identify the problematic process by running a [Process Monitor](#) and adding a filter. The condition to be met is if the **Result contains Sharing Violation** then **Include** the event.

0xC0000142

Message: Initialization of the dynamic link library failed. The process is terminating abnormally

Additional information for error resolution: If there is a problematic .dll file, you can use [regsvr32](#) to register the dll file and try again. A large number of problematic files could be a sign of an underlying issue that needs to be resolved before you can install the application.

Windows Management Instrumentation (WMI) errors

Error code	Error source	Error message
0x80041001	Windows Management Instrumentation (WMI)	WBEM_E_FAILED
0x80041009	Windows Management Instrumentation (WMI)	WBEM_E_NOT_AVAILABLE
0x8004100E	Windows Management Instrumentation (WMI)	WBEM_E_INVALID_NAMESPACE

General WMI troubleshooting tips

Problematic namespaces can typically be found in the [Configuration Manager log files](#) and the [WMI logging](#). WMI relies on Component Object Model (COM)/Distributed Component Object Model (DCOM), the registry, the file system, and Remote Procedure Call (RPC). DCOM registrations and permissions are critical for WMI operations to be successful. You can review DCOM configuration settings using [dcomconfig](#).

When troubleshooting WMI problems, typically you start by verifying that the needed namespaces, classes, and instances exists in the WMI repository and can be accessed.

Verify the namespace exists on the target first by running `wmimgmt.msc` from an elevated command prompt. When WMI Control launches:

1. Select Action then Properties.
2. Select the Security tab to see all the namespaces.
3. Navigate to the namespace in question.
4. Verify the namespace exists and review the security on the namespace.

To connect WMI Control to another computer:

1. Select Action then Connect to another computer.
2. Select the option for Another computer: then supply the name.

3. Select **Properties** to connect. The connection to the WMI repository on the remote computer doesn't occur until you select **Properties**.
4. Verify the namespace exists and review the security on the namespace.
5. You may also wish to try to connect with the IP address too to verify that you can connect.

Verify the namespace exists on the target and that you can query it properly. Run the Windows Management Instrument Tester from an elevated command prompt by typing in `wbemtest`. When the Windows Management Instrument Tester launches:

1. Select **Connect...**
2. Type in the problematic namespace such as `root\cimv2` or `root\ccm` and user credentials if needed. To connect to another machine, supply the name or the IP address such as `\Machine1\root\ccm` and credentials if needed.
3. Select **Enum Classes...** to verify you get classes listed for the problematic namespace.
4. Set the superclass info to **Recursive** and select **OK** to verify classes list for the problematic namespace.
5. Launch the object editor for one of the classes by double-clicking on it.
 - If you're using the `root\ccm` namespace, select a class that starts with "CCM_" such as `CCM_ClientIdentificationInformation`.
 - If you're using `root\cimv2`, choose one that starts with "Win32_" such as `Win32_BIOS`.
6. Select **Instances** to verify the instances of the selected class load. For some classes, it's ok if there aren't any instances, just make sure that the **Query Result** window states **Done**. Long running queries to list of instances or queries that never finish may indicate a problem.

Verify the repository:

1. From an elevated command prompt, run `winmgmt /verifyrepository`. Verifying is typically useful for invalid class errors especially if you had to recently recompile a .mof file using `mofcomp`.
2. If problems are found during verification, you can try to salvage using `winmgmt /salvagerepository`
3. Typically, you won't use `/resetrepository` unless it's truly needed and no other alternative exists. Some namespaces won't automatically rebuild and you'll need to either reinstall the software associated with the missing namespace or `mofcomp` the application's .mof files to rebuild them.

WMI resources:

- [Introduction to wbemtest](#)
- [Winmgmt service](#)
- [WMI Log Files](#)
- [Enable trace and debug logging for WMI events](#)
 - Ensure you change the default log size to cover your troubleshooting session.
 - Once you have finished troubleshooting, remember to disable the trace and debug logging.
- [Setting namespace security with the WMI Control](#)
- [WMI troubleshooting](#)
- [Ask The Performance Team: WMI ↗](#)

0x80041001

Message: WBEM_E_FAILED

Additional information for error resolution: WBEM_E_FAILED is a generic WMI failure error. The error can be caused by a number of things. The error will sometimes tell you which method or instance failed. You'll probably also see related log entries around the same time if you merge logs together based on similar function. For instance, if you see the error related to content for an application, you may want to merge together CAS.log, ContentTransferManager.log and DataTransfer.log. If the error happened on a site server not a client, you may want to review SMSProv.log for additional information. Use the [General WMI troubleshooting tips](#) to help identify the issue along with the application installation logs.

0x80041009

Message: WBEM_E_NOT_AVAILABLE

Additional information for error resolution: The resource, in many cases a remote machine, isn't currently available. Verify the device is online. Use the [General WMI troubleshooting tips](#) to help verify connectivity to WMI on the device.

0x8004100E

Message: WBEM_E_INVALID_NAMESPACE

Additional information for error resolution: The namespace specified could not be found. Verify the target computer can connect to WMI by following the [General WMI troubleshooting tips](#). Verify namespace specified exists.

Windows Update Agent errors

Error code	Error source	Error message
0x00240006	Windows Update Agent	The update to be installed is already installed on the system
0x80240017	Windows Update Agent	Operation was not performed because there are no applicable updates

General Windows Update Agent troubleshooting tips

The errors for the installation originated from the Windows Update Agent. In many cases, you can attempt to install these updates using the built-in software update management from Configuration Manager, Windows Update for Business, or Microsoft Update. In certain circumstances where it's not feasible to use your regular patching mechanism, the `.msu` package can be installed with the [Windows Update Standalone Installer \(wusa.exe\)](#) like an application. Use the [Windows Update logging](#) and [general troubleshooting tips](#) to help determine the cause of the issue.

0x00240006

Message: The update to be installed is already installed on the system

Additional information for error resolution: The update is already installed on the device.

0x80240017

Message: Operation was not performed because there are no applicable updates

Additional information for error resolution: The update isn't applicable to the device. Verify that the device meets the requirements of the update. In cases where a superseding update has been installed, it's very rare that the superseded update would be applicable to the device.

Troubleshoot the Microsoft Store for Business and Education integration with Configuration Manager

Article • 10/28/2022

This article provides key troubleshooting tips and fixes for some of the top issues that you may have with the Microsoft Store for Business and Education (MSfB) integration with Configuration Manager.

For more information about using the Microsoft Store for Business and Education with Configuration Manager, see [Manage apps from the Microsoft Store for Business and Education with Configuration Manager](#).

Monitor

Component status

In the Configuration Manager console, go to the **Monitoring** workspace, expand **System Status**, and select the **Component Status** node. Monitor status of the following components:

- SMS_BUSINESS_APP_PROCESS_MANAGER
- SMS_CLOUDCONNECTION

Sync status

In the Configuration Manager console, go to the **Administration** workspace, expand **Cloud Services**, and select the **Microsoft Store for Business** node. Check the **Last Sync Status** column.

View synchronized apps

In the Configuration Manager console, go to the **Software Library** workspace, expand **Application Management**, and select the **License Information for Store Apps** node.

Log files

WSfBSyncWorker.log

This log file is located on the service connection point, under `\Logs` in the Configuration Manager installation directory. It records information about the communication with the cloud service. This information includes metadata, icons, packages, and license file retrieval.

To change the log level, change the `LogLevel` value to `0` in the `HKLM\SOFTWARE\Microsoft\SMS\Tracing\SMS_CLOUDCONNECTION` registry key. For more information, see [Configure logging options](#).

SMS_CLOUDCONNECTION.log

This log file is located on the service connection point, under `\Logs` in the Configuration Manager installation directory. If the WSfBSyncWorker service isn't started, or repeatedly starts and stops, review the entries in this log file.

 **Note**

This log file is shared with other features.

BusinessAppProcessWorker.log

This log file is located on the site server for the top-level site in the hierarchy. It's under `\Logs` in the Configuration Manager installation directory. It records information about the following processes:

- Insert the metadata information synced by the BusinessAppProcessWorker component into the database
- Process files in `\InstallDir\inboxes\businessappprocess.box`

SMS_BUSINESS_APP_PROCESS_MANAGER.log

This log file is located on the site server for the top-level site in the hierarchy. It's under `\Logs` in the Configuration Manager installation directory. If the BusinessAppProcessWorker service isn't started, or repeatedly starts and stops, review the entries in this log file.

Last sync failed

When the last sync status is *failed*, start by reviewing the following log files to identify the symptom:

- WSfbSyncWorker.log
- SMS_CLOUDCONNECTION.log

Then look at one of the following sections for common issues:

- Authorization error
- The secret key is invalid
- Error getting application token
- Content location doesn't exist or incorrect permissions
- Error occurred making http request calling 'GET' method
- Cannot write more bytes to the buffer
- Online application download fails with 0x8024500c

Authorization error

Cause

This issue can occur if the configured Microsoft Entra application doesn't have permissions to manage the Microsoft Store for Business and Education for this tenant.

Workaround

1. Sign in as an administrator to the Microsoft Store for Business or Education portal.
2. Go to **Settings**, and select **Management tools**.
3. If the application isn't listed, select **Add a management tool**. Then search by name and select the Microsoft Entra application associated with the same ClientID as Configuration Manager.
4. If the status doesn't show **Active**, then select **Activate** in the **Action** section.
5. In the Configuration Manager console, go to the **Administration** workspace, expand **Cloud Services**, and select the **Microsoft Store for Business** node. Synchronize with the store, or wait for the next sync interval to occur.

💡 Tip

To find the ClientID in Configuration Manager:

1. In the Configuration Manager console, go to the **Administration** workspace, expand **Cloud Services**, and select the **Microsoft Entra Tenants** node.

2. Select the tenant that you use for the Microsoft Store for Business and Education integration.
3. In the results pane, find the matching application, and look at the **Client ID** column.

The secret key is invalid

Cause

This issue can occur if the secret key has expired on the Microsoft Entra app for the Microsoft Store for Business and Education configuration.

Resolution

Renew the secret key for the Microsoft Entra application. For more information, see [Renew secret key](#).

Error getting application token

Cause

This issue can occur if the connected app no longer exists in Microsoft Entra ID.

Resolution

Delete and recreate the connection to the Microsoft Store for Business and Education.

1. In the Configuration Manager console, go to the **Administration** workspace, expand **Cloud Services**, and select the **Microsoft Store for Business** node.
2. Select the existing connection.
3. Select **Delete** in the ribbon.

Then recreate the connection. For more information, see the following articles:

- [Configure Azure Services](#)
- [Set up Microsoft Store for Business and Education synchronization](#)

Content location doesn't exist or incorrect permissions

Cause

When you set up the Microsoft Store for Business and Education connection, you specify a network share for storing synchronized content. This issue can occur if this share doesn't exist or has incorrect permissions. The computer account for the service connection point should be the owner of this directory and any sub-directories. If it isn't, you'll see an error similar to the following error:

```
Failed to download package d788cc1b-ab00-bb5f-1548-f2dfe717583b-X86-Arm for product 9WZDNCRFJ3PS\0015.  
System.IO.IOException: This security ID may not be assigned as the owner of this object.
```

To see the location that you configured:

1. In the Configuration Manager console, go to the **Administration** workspace, expand **Cloud Services**, and select the **Microsoft Store for Business** node.
2. Select the account and open its **Properties**.
3. Switch to the **Configuration** tab. The **Location** setting shows the network path to store application content downloaded from the Microsoft Store for Business and Education.

Workaround

1. If it doesn't already exist, create the share.
2. Check NTFS permissions on the folder, and the permissions on the network share. Grant the computer account of the service connection point **Read** and **Write** permissions.

If you want to reconfigure the location, delete and recreate the connection with the new content location.

Error occurred making http request calling 'GET' method

Cause

This issue can occur if the sync of applications from the store took so long that the content URL expired.

Workaround

Retry the sync process

1. In the Configuration Manager console, go to the **Administration** workspace, expand **Cloud Services**, and select the **Microsoft Store for Business** node.
2. Select the connection. In the ribbon, select **Sync** from **Microsoft Store for Business**.

With each time, it should continue further. It may take several retries depending on the following factors:

- The number of offline applications
- The size of the packages
- The network speed

With each attempt, you should see the error fewer times. If the number of errors doesn't reduce, there's another issue.

Cannot write more bytes to the buffer

Cause

This issue can occur if the application's package is larger than 500 MB. Configuration Manager only supports automatic synchronization of offline applications with packages less than 500 MB.

Workaround

You can't automatically sync these apps, but you can download the content, and manually create the application:

1. Get the failing application ID from the following line in **WSfbSynWorker.log**:

```
Error(s) syncing or downloading application <ApplicationID> from the Microsoft  
Store for Business.
```

2. Sign in as an administrator to the Microsoft Store for Business or Education portal. Find the page for this application.

 Tip

The URL for the page is similar to: <https://businessstore.microsoft.com/en-us/store/p/app/ApplicationID>

- a. Select **Offline**, if it isn't already selected. Then select **Manage**.
 - b. Create a separate folder on your application content share for all supported platforms.
 - c. Download the package to the package folder.
 - d. Download the encoded license file as a **.bin** file to the package folder.
 - e. Download all required frameworks to the package folder.
3. In the Configuration Manager console, go to the **Software Library** workspace, expand **Application Management**, and select the **Applications** node.
 4. [Create an application](#), manually specifying the application information.
 - a. Create a deployment type for each supported platform that you previously downloaded.
 - b. Type: **Windows app package (*.appx, *.appxbundle)**
 - c. Specify the appx/appxbundle for the actual app package, not a required dependency package.

Confirm the following details on the final **Import Information** page:

- **License file:** Specifies the **.bin** file. This license file is required for offline apps.
- **Windows app dependencies:** Verify that all of the required dependencies are downloaded for this package.

Online application download fails with 0x8024500c

Cause

An 0x8024500c error during download is typically caused by the **Do not connect to any Windows Update Internet locations** group policy that blocks Windows Update access.

Workaround

Don't enable the **Do not connect to any Windows Update Internet locations** group policy object.

Sync doesn't run

This section covers the following sync issues:

- You manually start the sync process, but it doesn't run
- The site doesn't automatically sync each day

Start by reviewing the following [log files](#) to identify the symptom:

- BusinessAppProcessWorker.log
- SMS_BUSINESS_APP_PROCESS_MANAGER.log
- WsfbSyncWorker.log
- SMS_CLOUDCONNECTION.log

Then look at one of the following sections for common issues:

- [Manual sync doesn't start](#)
- [Automatic daily sync doesn't run and "shutting down # workers" error in SMS_BUSINESS_APP_PROCESS_MANAGER.log](#)

Manual sync doesn't start

Cause

This issue can occur if you start a sync less than 10 minutes after the previous sync. You can't sync more frequently than every 10 minutes.

Resolution

Wait for at least 10 minutes before starting another sync.

Automatic daily sync doesn't run and "shutting down # workers" error in SMS_BUSINESS_APP_PROCESS_MANAGER.log

Cause

This issue can occur if the SMS_BUSINESS_APP_PROCESS_MANAGER component stops the WsfbSyncWorker thread. The error may specify either 2 or 4 workers.

Workaround

Restart the **SMS_EXECUTIVE** service.

If you're not able to restart that main service, stop both components with MSfB workers, and then start both:

1. Open the Windows registry on the server that runs the service connection point

2. Go to

`HKLM\SOFTWARE\Microsoft\SMS\COMPONENTS\SMS_EXECUTIVE\Threads\SMS_CLOUDCONNECTI
ON`

a. Set Requested Operation to **Stop**.

b. Refresh to verify Current State = **Stopped**.

3. Go to

`HKLM\SOFTWARE\Microsoft\SMS\COMPONENTS\SMS_EXECUTIVE\Threads\SMS_BUSINESS_APP_
PROCESS_MANAGER`

a. Set Requested Operation to **Stop**.

b. Refresh to verify Current State = **Stopped**.

4. In **SMS_CLOUDCONNECTION**, set Requested Operation to **Start**.

5. In **SMS_BUSINESS_APP_PROCESS_MANAGER**, set Requested Operation to **Start**.

Language-related issues

This section includes the following common issues:

- [Language selection changes aren't applied](#)
- [Not all selected languages are present for all license information](#)

Language selection changes aren't applied

Cause

This issue can occur if the language selection is cached, and isn't cleared after the property values are changed.

Workaround

To resolve this problem, restart the **SMS_Executive** service.

Not all selected languages are present for all license information

Cause

This issue can occur if the Microsoft Store for Business and Education application's license information doesn't contain localized data for the specified language.

Workaround

Manually add any missing languages for created applications.

Offline applications

This section includes the following common issues:

- [Fail to create offline application because content can't be verified](#)
- [Fail to install application created from offline license information](#)

Fail to create offline application because content can't be verified

Cause

This issue can occur if the synchronized content for the offline application is corrupt or modified.

Workaround

Start a new sync. When the sync completes, it should verify and download any incorrect content files.

Fail to install application created from offline license information

Cause

This issue can occur if you deploy the application to a client running a version of Windows 10 earlier than version 1511. Offline licensed apps from the Microsoft Store for Business and Education are only supported on Windows 10 version 1511 and later.

Resolution

Install the latest version of Windows 10.

Next steps

To find additional help, see [Find help for using Configuration Manager](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Troubleshoot Package Conversion Manager

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

Use the information in this article to help you troubleshoot problems when using Package Conversion Manager.

SMS Provider

Package Conversion Manager uses the SMS Provider. For more information, see [Plan for the SMS Provider](#).

If the SMS Provider isn't working properly, the Configuration Manager console including the Package Conversion Manager doesn't work.

Package readiness

Before converting a package to an application, analyze the package using the Package Conversion Manager **Analyze** function. After the analysis, add the **Readiness** column in the **Packages** node of the Configuration Manager console. The list of packages displays one of the following readiness states of the analyzed package:

- **Automatic:** The package can be directly converted using the **Convert** function.

(!) Note

An automatic conversion doesn't convert WQL queries into application requirements. Use the **Fix and Convert** process to convert these queries.

- **Manual:** The package needs some additions or changes before you can convert it using the **Fix and Convert** function.
- **Not Applicable:** The package isn't suitable for conversion. Either correct any problems with the package, or continue to deploy it as a package.
- **Error:** The package contains errors. Manually correct these errors before you can analyze and convert it.

The details pane of the **Packages** node in the Configuration Manager console shows any readiness issues. Select a package, and then select the **Summary** tab in the details pane.

Log files

Enable logging

When you enable logging for Package Conversion Manager, it logs all of its actions, exceptions, and errors.

To enable logging for this component in the Configuration Manager, modify **Microsoft.ConfigurationManagement.exe.Config**. By default, this configuration file is located in the following path:

```
C:\Program Files (x86)\Microsoft Endpoint Manager\AdminConsole\bin\Microsoft.ConfigurationManagement.exe.config
```

ⓘ Important

Starting in version 1910, this path changed to use the **Microsoft Endpoint Manager** folder. Make sure you don't use an older version of the file that might exist in another folder.

Insert the following **switches** and **trace** XML elements in the **system.diagnostics** element after the last **sources** element:

XML

```
</sources>

<switches>
    <add name="PcmLogging" value="3"/>
</switches>
<trace autoflush="true" indentsize="4">
    <listeners>
        <add name="PcmTraceListener"
type="Microsoft.ConfigurationManagement.UserCentric.Logging.RolloverLogTrace
Listener, Microsoft.ConfigurationManagement.UserCentric.Logging"
initializeData="%UserProfile%\AppData\Local\Temp\PcmTrace.log"/>
    </listeners>
</trace>

</system.diagnostics>
```

This sample uses the file **PCMTrace.log**. This log is on the computer running the Configuration Manager console in the following path:

```
%UserProfile%\AppData\Local\Temp
```

To configure the level of detail, change the **PcmLogging** trace switch setting. Set the this value to four levels of detail, from least detailed (1) to most detailed (4).

SMSProv.log

In some situations, information relevant to troubleshooting the package conversion process is in the **SMSProv.log** file. This file captures information from the Configuration Manager SMS Provider.

By default, this log file is located on the Configuration Manager site server at the following path:

```
C:\Program Files\Microsoft Configuration Manager\Logs
```

If you see one of the following error messages, the **SMSProv.log** file may contain relevant troubleshooting information:

- The SMS Provider reported an error
- Generic Failure

These error messages typically indicate that an error occurred on the site server, and that the error information wasn't sent to the Configuration Manager console.

For more information, see [Technical reference for Package Conversion Manager error messages](#).

Changing package attributes after analysis

After you analyze a package and it has a readiness state of **Automatic** or **Manual**, the conversion process might fail if you change any of the relevant attributes.

For example, you analyze a package and its readiness state is **Automatic**. Then you add another program to the package. The package conversion might fail.

If you need to make changes to a package after analysis, rerun analysis before conversion.

See also

Technical reference for Package Conversion Manager error messages

Technical reference for Package Conversion Manager error messages

Article • 10/04/2022

Applies to: Configuration Manager (current branch)

This article describes the error messages that Package Conversion Manager displays. It also includes the possible causes of the error, and methods to correct the error. Package Conversion Manager logs error messages in **PCMTrace.log**. For more information, including how to control the verbosity level, see [Log files](#).

Application creation failed with the following exception

The specified exception occurred during the submission of the application object to the Configuration Manager server.

Check your permissions in Configuration Manager, validate your connectivity, and then retry. If those actions don't fix the problem, examine the **PCMtrace.log** file (verbosity level 4) and **SMSProv.log**.

Conversion Error – APPLIES TO A PACKAGE TRANSFORM STATUS

A general exception occurred during the conversion of the package. Look in the **PCMtrace.log** file (verbosity level 4).

Check the user permissions for the network share (package data source), validate your connectivity, and then retry. If those actions don't fix the problem, examine the **PCMtrace.log** file (verbosity level 4).

Did not find a converted package and its resultant application in the workflow outputs

The application (converted package/program) was deleted.

Modify the dependent package/program to ensure that the dependent package/program exists.

Objects were not created successfully

There are several possible causes.

Check your permissions in Configuration Manager, validate your connectivity, and then retry. If those actions don't fix the problem, examine the **PCMtrace.log** file (verbosity level 4) and the **SMSProv.log** file.

Please close the wizard and resolve any issues with the selected package. See PCMTrace.Log for more details

There are several possible causes.

Check your permissions in Configuration Manager, validate your connectivity, and then retry. If those actions don't fix the problem, examine the **PCMtrace.log** file (verbosity level 4) and the **SMSProv.log** file.

Some Deployment Types are missing Detection Methods. All Deployment Types must have Detection Methods

Detection methods are missing from the program.

Add one or more detection methods during the **Fix and Convert** process.

There was an error preparing the package for conversion

There are several possible causes.

Check your permissions in Configuration Manager, validate your connectivity, and then retry. If those actions don't fix the problem, examine the **PCMtrace.log** file (verbosity level 4) and the **SMSProv.log** file.