Recast

Manage Packages

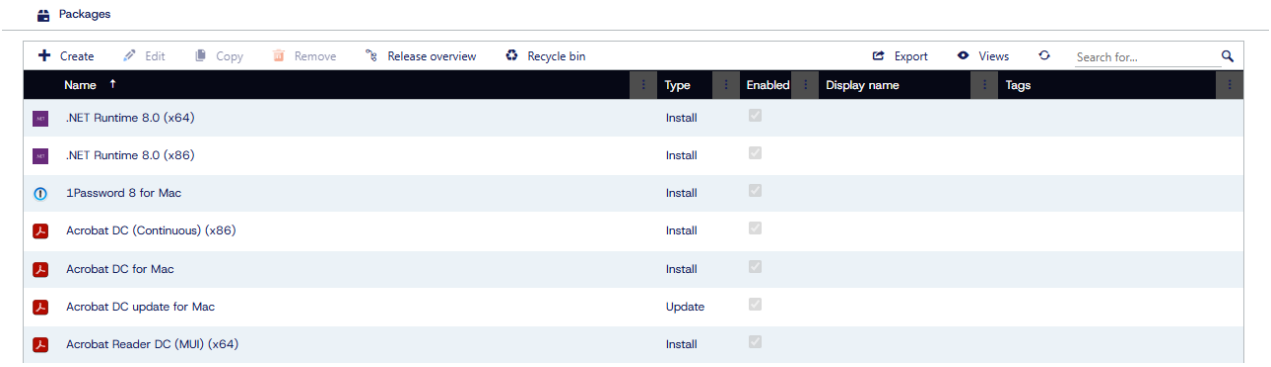
Last Modified on 2026-11-02

View Packages

The Application Workspace **Packages** page displays information about all your packages such as the type of package, whether it is enabled, its display name, and any tags applied to the package.

From this page, you can:

- Create a new package or edit an existing one
- Copy an existing package, as a starting point for a creating a new one
- View the Release overview
- Review the contents of the Recycle bin



View the Release Overview

The **Release overview** can help you determine which packages need to be updated. It displays the name and type of a package, if it is enabled or managed, the version number (if they exist) corresponding with all release stages (Development, Test, Acceptance and Production), and the source of the app (the Application Workspace Setup Store, for example).

You can access the release overview by clicking on the **Release overview** button in the table toolbar.

> The release overview for packages requires a Release and Patch Management license.

Review Recycle Bin Contents

When you remove a package, it will be moved to the recycle bin and will remain here until the Clean Packages scheduled task is executed or until you manually delete it.

You can also do the following:

Restore one or more packages you previously selected. Whenever Application Workspace encounters a package with the same name, you will be prompted to rename the package to avoid duplicates. When a package is successfully restored, it is moved back to the list of packages and can be assigned and edited like any other package within the list.

Remove one or more packages you previously selected from the system. This action is permanent; there is no way to restore them afterwards.

Recast

Package Types

All these package types come with predefined basic settings, except for **Custom package**, where the settings are blank.

Install MSI – Install an .MSI package by uploading it or providing Application Workspace with the network credentials to access it.

Install Mac App – Install an app via a .PKG file on a macOS platform.

Local application – Configure a launch action for an existing local app from the user's device.

Modify group membership – Change the group memberships for Entra ID and Active Directory.

- For Entra ID, you'll first need to go to **Identity Sources** > **Settings** > **Synchronization** > **Modifications** and select 'Group membership'.
- For LDAP, you'll first need to go to **Identity Sources** > **Settings** > **Synchronization** > **Modifications** and select 'All'.

Remote application – Deploy an app that launches a Remote desktop session.

Remote control – Deploy a remote package to the user device and launch a remote control session for that device via your desired remote desktop software application.

Web application – Make a link available to your users, that launches when they click on the Workspace app or when they log in for example.

Custom package – Unlike the other types of packages that have some settings preconfigure for you to cover the basics, a custom package represents a complete blank template.

> The **Superseded by products** information that appears in the **Create new package** window shows all the product versions within Application Workspace Setup Store that supersede the currently selected version.

View/Edit Package Options

From the Application Workspace **Packages** page, you can view and/or edit the options associated with an individual package by double-clicking on its entry.

> If the package is managed by a connector, some options may be unavailable.

Overview

The Overview tab display basic package information.

Type – This information is read-only, with one exception: the **Remote application** package type has 'Remote' auto filled for this field and must not be changed.
Priority – Determines the execution order when multiple packages are assigned to an event (on a user or device

entitlement); the order of execution is from low to high.
Agent Required – Determines if the package is available only when a Application Workspace Agent is running.
Offline – Determines if the package is available for users that use the offline mode of the Application Workspace Agent; this causes the package content to be distributed on login/refresh, to ensure that the package is available for launch/installation actions. This is useful, for example, in situations where the user wants to start an application but the device is in a remote location with no internet connection, or the company policy forbids connecting to unknown access points.
User auto launchable – When enabled, users can choose to start this package on logon by enabling the **Autolaunch** option in the context menu of the app. This feature is available only with events enabled; events are enabled by default.

Events

Displays all events that are associated with the package.
The following columns are available:

Type – The type of the event associated with the package.
Status – Shows whether or no the event was completed successfully.
Identity – The user that triggered the event. For example, the user who initiated a package launch.
Device – The device the user was logged on when the event was triggered.
Package – The name of the package associated with the event. This column is hidden by default.
Time – The date and time the event was triggered.

Auditing

View a comprehensive log of changes to this package, displaying the identity behind each modification.
This screen is available only if the auditing is enabled in the Database Event Collector. For more information, see Auditing.

Entitlements

Allows you to assign identities like users, user collections, groups, devices, device collections or contexts to the current package. This grants users or devices access to an application.
To view or edit the details of an identity, double-click its entry.

The **Edit entitlement** window will open with the following options:

Overview tab

Identity – The display name of the existing identity you selected for this entitlement.
Stage – The release stage used for this identity; this feature is only available for Release and Patch Management license.
Publish – The package can be published in one of these ways:

- **Hidden** – The package is available to the user but is hidden. This can be useful for example if you only want to start this application with an event.
- **Catalog** – Publishes the package to **Catalog**, where users can get/request the application if they need it.
- **Workspace** – Publishes the package to **Workspace**. The user can remove the package which will automatically return to **Catalog**.
- **Forced** – The package is pushed forcefully on **Workspace** and cannot be removed by the user.

Native Icons – Select where the native icon of the package should be displayed for Windows or Mac OS. For it to work, an Agent must be running on devices.

- **Desktop** – The icon will be displayed on the user's desktop, on a Windows OS.
- **Start Menu** – The icon will be displayed in the user's Start Menu, on a Windows OS.
- **Taskbar** – The icon will be displayed in the user's Taskbar, on a Windows OS. Issues have been reported with Windows version 10 1903 or higher; see known issues for more details.
- **macOS Desktop** – The icon will be displayed on the user's desktop, on a macOS.

Approver – Gives the identity the rights to request and approve an application from **Catalog**. Works only if an app is published in **Catalog**, from where the user is able to request it following the steps described in Catalog.

> The **Managed** column on the **Entitlements** tab denotes that the entitlement comes from a connector, configured as a synchronized connector that manages the package, like Application Workspace, Remote Desktop and Omnissa Horizon. It cannot be modified.

Events tab

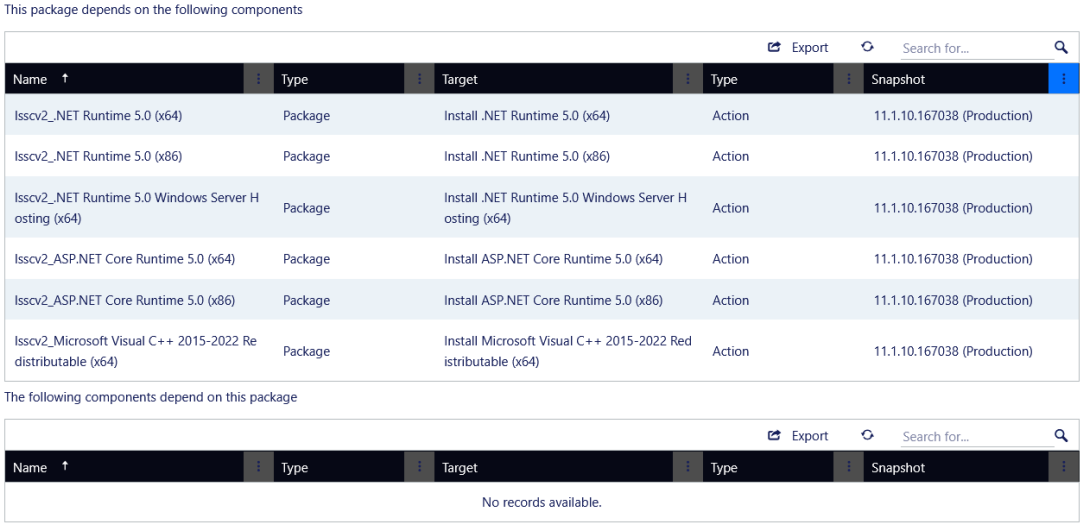
You can configure an event to trigger an action on this package. To add an event, click the **Add** button; to edit an existing one, double-click its entry.
The following triggers are available:

Application Workspace Logon – Triggers when a user logs in to the Application Workspace.
Refresh – Triggers when a refresh is triggered.
Session connect – Triggers when a remote desktop session is connected/reconnected.
Session disconnect – Triggers when a remote desktop session is disconnected/interrupted.
Session lock – Triggers when a desktop session is locked.
Session unlock – Triggers when a desktop session is unlocked.
Agent startup* – Triggers when the Agent service is started.
Device startup* – Triggers when the device is started. This option does not work for devices with Windows OS that have the Fast startup or Hibernate enabled.
Device shutdown* – Triggers when the device is shutdown. This option does not work for devices with Windows OS that have the Fast startup or Hibernate enabled.
User login – Triggers when a user logs in to the device.
User logout – Trigger when a user logs out of the device.
Network change – Triggers when a network (interface/adapter) change occurs. The changes are grouped together when there is less than 5 seconds between them and will happen over a maximum time frame of 20 seconds.

* This event is available only on device assigned entitlements.

Dependencies

Lists all the components this package depends on and components that depend on this package. When you double-click an entry, the referred action is displayed.
NOTE: Only package type actions can create dependencies e.g.: distribute, install, launch, repair, uninstall.



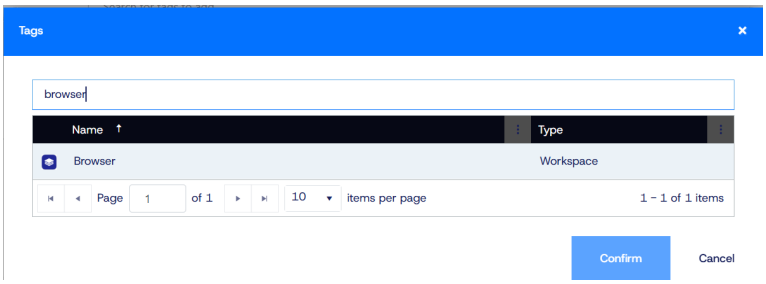
Recast

Tags

View, remove or add tags to your package.

To add a tag, start typing in the Search for the desired tag and select it from the results lists.

TIP: Alternatively, you can select the ellipsis (...) to the right of the Search to view all the tags available.



To view the details of a tag or edit it, double-click its entry.

> The source of a managed tag is a system certificate and it cannot be changed.

Catalog section

The catalog section provides basic options to configure the manner in which the application is displayed for the user in the **Catalog** tab of Application Workspace.

Catalog Overview

Display name – The name displayed in the **Catalog** for the user. By default, it is the name you give to the app when you created it, but you can write a different name in this field.
Featured on the catalog – When checked, the package will be displayed in the Featured section of the **Catalog**, at top of the tab. See Catalog for more details.
Excerpt – A shorter version of the application description, displayed on the application tiles.

Description – The longer version of the application description, displayed in the catalog information screen.
See Catalog for more details.
Website – Provide users a website associated with the application. This can be the official website or for example a knowledge base article within the organization. The **Website** button will be displayed on the application tiles and catalog information screen. See the Netflix example in the image above.

Catalog Reviews

Manage reviews written by users about the application. You can moderate and delete reviews.

Catalog Media

The media screen allows you to add images which are displayed in the user catalog. You can choose to display the image selected as:

- *Banner* displayed only for apps that have the **Featured on the catalog** option checked.
- *Screenshot* displayed in the catalog information screen of the app.
- *YouTube Videos* displayed in the catalog information screen of the app

Recast

For YouTube videos to load properly, you'll need to add the following URL to CORS and CSP in the Security Settings: `https://*.youtube-nocookie.com`

Catalog FAQ

The FAQ allows you to add frequently asked questions to the user catalog.
When you click on the **Create question** button, a dialog pops-up which allows you add a question and answer. You even have some basic formatting options at your disposal for the answer.

Catalog Categories

Displays a list of categories associated with the current package.
View, remove or add categories to your package. To add a category, simply start typing in the lookup field for the desired category and select it from the results lists. Or select the browse button ... at the right of the lookup field to open the **Categories** dialog box and view all the categories available. To view the details of a category or edit it, double-click its entry.
For more information see Categories.

Catalog Licenses

Displays a list of product licenses associated with the current package.
View, remove or add licenses to your package.

To add a license, start typing in the Search for the desired license and select it from the results lists.

TIP: Alternatively, select the ellipsis (...) to the right of the Search to view all the licenses available.

To view the details of a license or edit it, double-click its entry.
For more information see Licenses.

Releases section

An individual package can have different package snapshots in different stages. Package staging allows you to test updates in multiple stages of the application lifecycle before its deployed to all users in the Production stage. The available stages are:

- Development – This is the stage in which you can update the package. Assigning this stage to users is not recommended as any change in the package will be live for the assigned users right away.
- Test – This is the stage you can assign to users for test purposes. Usually, test users test the package on a technical level. (e.g. check if the package actions work as expected)
- Acceptance – This stage is used for acceptance purposes. Assign users who can accept the package prior to updating the application to production in this stage. Usually, acceptance users test the package on a functional level. (e.g. check if the application works as expected)
- Production – This is the stage most widely used in production environments. After a package is changed within Development, tested within Test and accepted in Acceptance by the users, you can place the package in production.

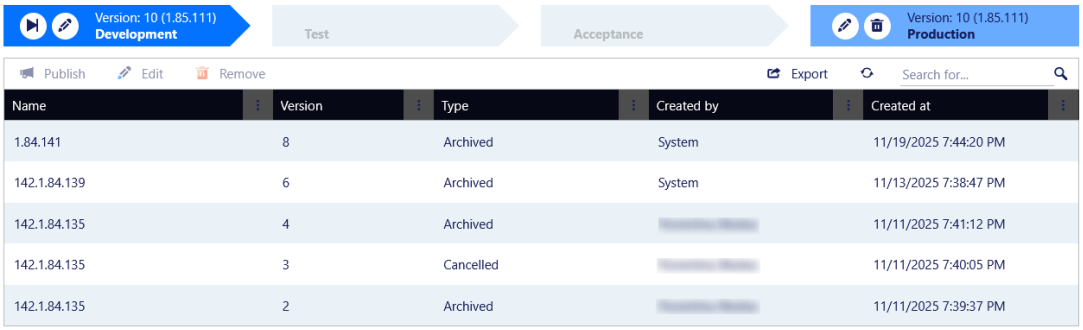
The Test and Acceptance stages are available with a Release and Patch Management license.
Packages can be modified only in the Development stage.
When a package in the development stage is being distributed/installed/launched then both distribute and install action sets will be executed forcefully every time the package is triggered. The configured action set frequencies will be ignored.

Recast

Releases Overview

Displays the release dashboard and a list of cancelled and archived versions of a package.

- Cancelled stage – a stage is cancelled when:
  - you overwrite the version of the stage with a package version from another stage
  - you skip it by moving a package version from a lower stage to a stage higher than the current one
  - you delete the stage by clicking on the Bin icon on the stage
- Archived stage – only a version from the Production stage can become archived, after you overwrite it with a new version.



Each of the snapshots listed in the table can be recovered at any time; what's more, you can choose to publish it to any stage. Just select the desired snapshot, then click **Publish** in the table toolbar and in the **Publish "Release"** dialog box that opens, select the stage to which you want to publish the snapshot and change the name and description of the release if needed.

Release Shortcuts

Shortcuts allows you to define multiple icons and names for a single package that are published within the **Workspace**, but are listed as a single item in the **Catalog**. Any of the shortcuts can be disabled, thus hiding them when the application is added to the **Workspace**.
A "primary" shortcut is always available on a package and can't be removed. It contains the default icon displayed to the user, which you can overwrite, and allows you to override the display name configured in **Catalog** section > **Overview**.

Release Variables

Displays all the system variables configured in **Workspace** > **Variables**. Variables give you the tools to be as dynamic as possible as they can currently be defined on different levels. The inheritance flows top-down and supports local override. This means a variable defined on system level, can be used in contexts and packages. That same variable could be overridden on for example package level, to provide dynamic customization based on the entity.
In the Development stage you can override any of the existing variables by clicking on them or you can create new variables by clicking on the **Create variable button** in the table toolbar.
For a variable named 'MY_CUSTOM_VAR' you must use the following notation in fields that support variables: `${MY_CUSTOM_VAR}`
For more information about system variables see Variables.

Release Filters

Filters allow you to define when a package is available for a user/device based on certain requirements. For example, only enable a package if a certain file is available on the device that is used to access the Application Workspace. Filters are grouped by filter sets, and you can use the 'AND' or 'OR' operators between sets of filters and within sets.
For detailed information about all the filters available for a package see Filters.

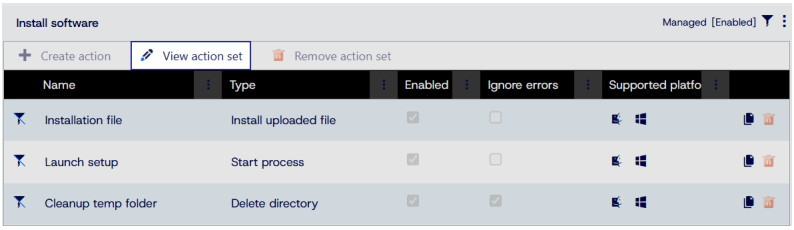
Release Actions

Recast

On the Releases **Actions** tab, you can define the behaviour of the package by creating actions. Actions are grouped by action sets, and you can use the 'AND' or 'OR' operators between sets of actions and within sets.

To view an existing action set:

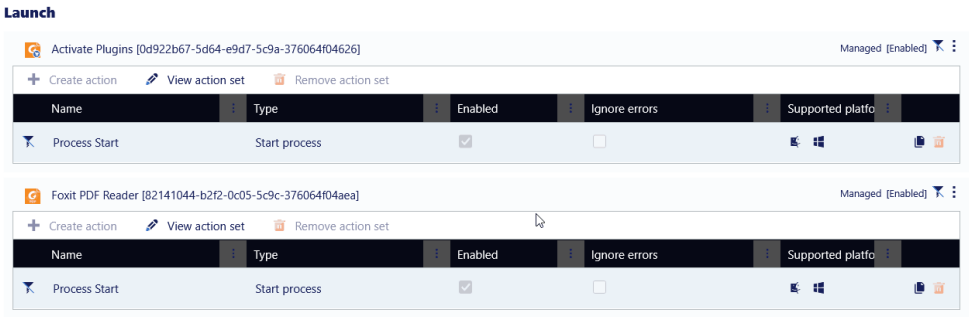
1. On the **Actions** tab for a selected package, click **View action set**.



2. In the **View action set window** that opens, verify Overview information and/or Filters.

For detailed information about all the actions available for a package see Actions.

When the type of action set is the same, you can change the order of execution of actions sets by clicking on the title bar of the action set and dragging and dropping it in the new location.



To create a new action for a package:

1. On the **Actions** tab for a selected package, click **Create action**.

2. On the **Overview** tab, select Type, Repair, Shortcut, and Frequency options.

Overview tab

Type – Determines how actions are executed:

- Distribute – Is used to download all content that is needed for installation on devices and is executed only once per device
- Install – Is used to install the package on devices. Use the frequency option to specify when an installation is required
- Launch – Is executed each time the user clicks on the smart icon of the package
- Repair – Is executed when the user clicks on **Repair** in the Context menu of the app, in Workspace

  The **Repair** option becomes visible to a user, only after they click on an icon that implies executing an action on their device, e.g. copying files or launching an .exe file (meaning for a package with a Distribute, Install or Repair action set configured). It is not displayed for apps that open in the browser.
  If no Repair action set is defined in case of a package containing Distribute or Install action sets, the install actions will be relaunched.
- Uninstall – Executed only when it's triggered by an uninstall event configured for the identity entitled or triggered by a different package.

Shortcut – Shortcut selection is available only for a Launch action set.
Process – Determines the order the actions within the action set are executed:

- Sequential – The actions in the action set are executed in order, stopping at the first error
- Sequential, ignore errors – The actions in the action set are executed in order, ignoring errors
- Stop at first effective action – The first effective action in the action set will be executed while the rest will be ignored
- Stop at first successful action – The first successful action in the action set will be executed while the rest will be ignored

Frequency – Depending on the action set type there will be different options:

- Always – the action will always be executed
- Once per user – the action will be executed only once per user
- Once per device – the action will be executed only once per device

Filters tab

Here you can define conditions which determine if the actions within the set will be executed.Learn more about Filters.

To edit action settings:

1. On the **Actions** tab for a selected package, double–click on an action.

2. In the **Edit action** window that opens, edit settings on the **General** and/or **Advanced** tabs. The elements on the **General** and **Advanced** tabs vary according to the type of the action. See the full list of Actions.

3. On the **Filters** tab, set the conditions under which the action should run (optional). See the full list of Filters.

Create managed package with Setup Store connector

In this example using MongoDB Community Edition 6.0, we'll create a managed package in Application Workspace via the Application Workspace Setup Store.

To create a managed package:

1. On the **Connectors** page, select the Application Workspace Setup Store connector.

2. Click **Resources** in the table toolbar.

3. Search for and select the product – in this case, MongoDB Community Edition 6.0.

4. Click **Create package** in the table toolbar.

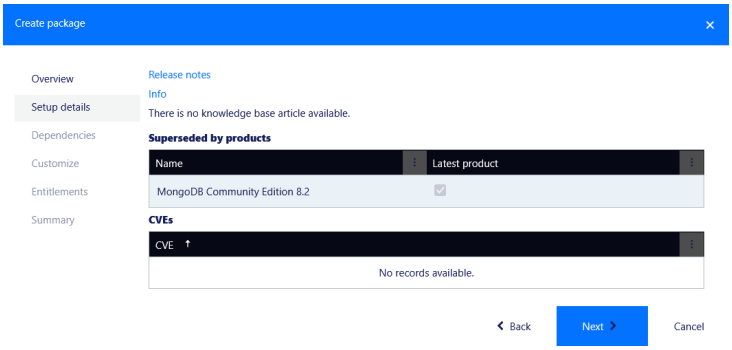
5. In the **Create package** window that opens:

On the **Overview** tab

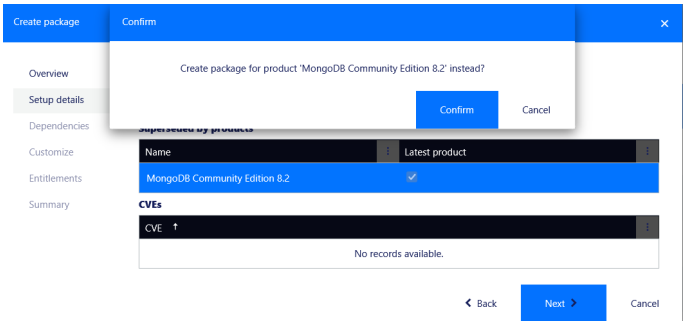
- Enter a name for your new package or keep the default name.
- As the **Package type**, select 'Managed'. Managed means that every time MongoDB releases a new version for this product, the Application Workspace Setup Store connector updates the package with the latest version. For more details about package types, see Connectors.
- In **Publish**, select the stage to which you want to publish the new package. For more details, see Package
- Leave the **Update existing packages** unchecked. If this option is selected, it will overwrite an existing package.

On the **Setup details** tab

- Access links to the release notes and product description on the developer's official website, plus links to Recast Help Center articles
- View virus scan scores and CVE reports, in case any vulnerabilities and exposures exist for this resource.
- In the **Superseded by products** table, view all the product versions within Application Workspace Setup Store that supersede the currently selected version.

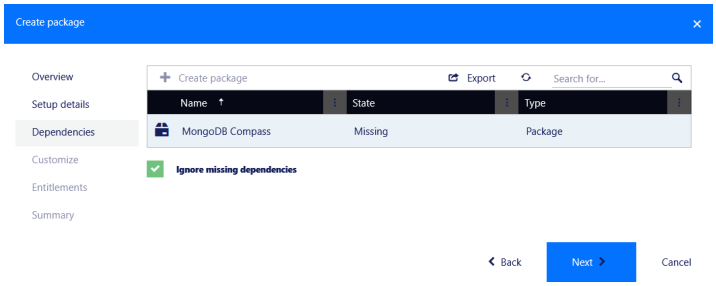


If you decide that you want to install one of the versions marked as superseded in this table, just double-click on it, and then click **Confirm**. It will open a new **Create package** dialog box for this version.

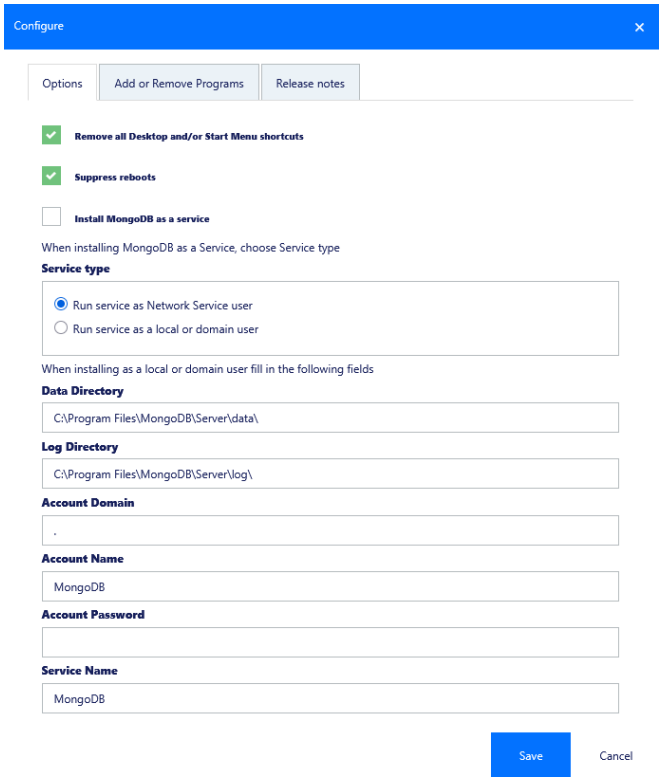


On the **Dependencies** tab, Application Workspace lists all prerequisite software components and automatically detects if they are installed or not on your local system. You can easily create a package for each uninstalled component right from the dialog box by double-clicking it or by selecting it and clicking **Create package** in the table toolbar. It opens a new Create package dialog box where you configure the new package.

- Select **Ignore missing dependencies** if you do not want to create managed packages for connected dependencies.



On the **Customize** tab, access the customization wizard for additional settings offered by the developer.



On the **Entitlements** tab, add the identity (like contexts, device collections, devices, groups, users and user collections) to which you wish to publish the newly created package.

On the **Summary** tab, leave the **Modify package after creation** enabled if you want to open the newly created package and further configure it.

> **Catalog information prefilled**
> If you configure the managed package to be published in the user's catalog, the following sections come prefilled: excerpt, description, media, details and website.

Further reading

CVE, Then and Now: Overview and Practical Integration with Application Workspace

---

Update managed package from Setup Store

You can update a managed package manually or automatically.

In this article, we'll use Mozilla Firefox as an example of updating a managed package created from the Application Workspace Setup Store. Application Workspace Setup Store is configured with the on-demand synchronization method, the only one supported.

To manually update the managed package:

1. Open the Mozilla Firefox package and navigate to **Overview** > **Updates**.

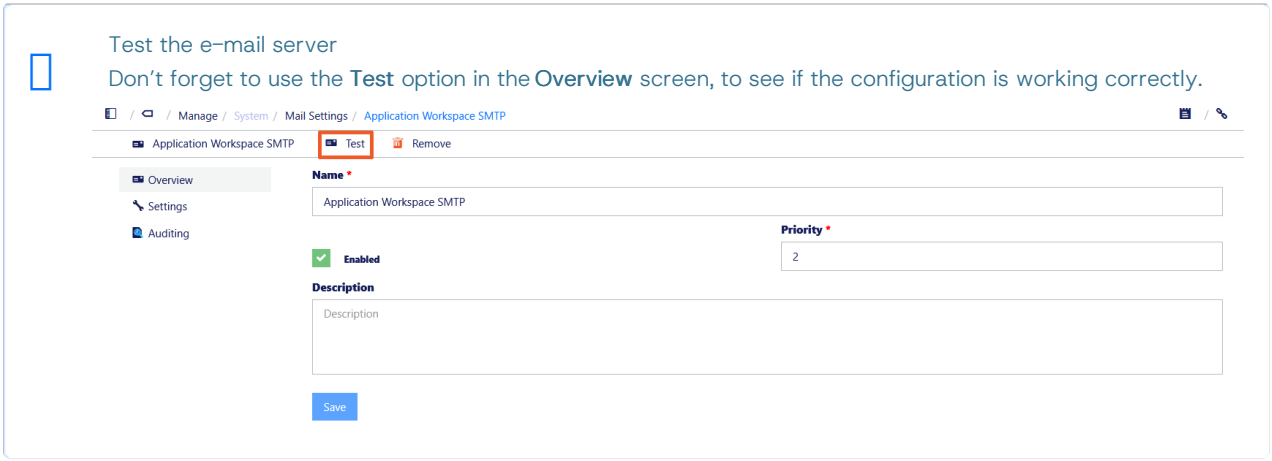
2. Select the new version of Mozilla Firefox and click **Apply update**.

To update the managed package automatically:

Configure the email settings

To receive e-mail notifications within Application Workspace, you first have to add an email server in **Manage** > **Mail Settings**.

Click **Create** and configure all the necessary settings for this email server. This email server is used by the scheduled tasks to send email messages. For more information, see Mail Settings.



Create a scheduled task

1. Navigate to **Manage** > **Automation** > **Scheduled Tasks**.

2. Click **Create**.

3. In the **Create scheduled task** that opens, select the 'Synchronize connector' type.

4. Under **Settings**:
   - Select an existing Application Workspace Setup Store.
   - Select **Download and apply updates**.
   - Enable **Send detailed per package notifications**. By using this option you will receive, per managed package, a very detailed email message containing information such as release notes, CVE, CVSS information (if applicable), and the virus scan score. This information is important for risk mitigation.
   - Select to which DTAP stage you want to deploy the updated package. Note that the default option represents the stage configured in the **Releases** screen of the connector.

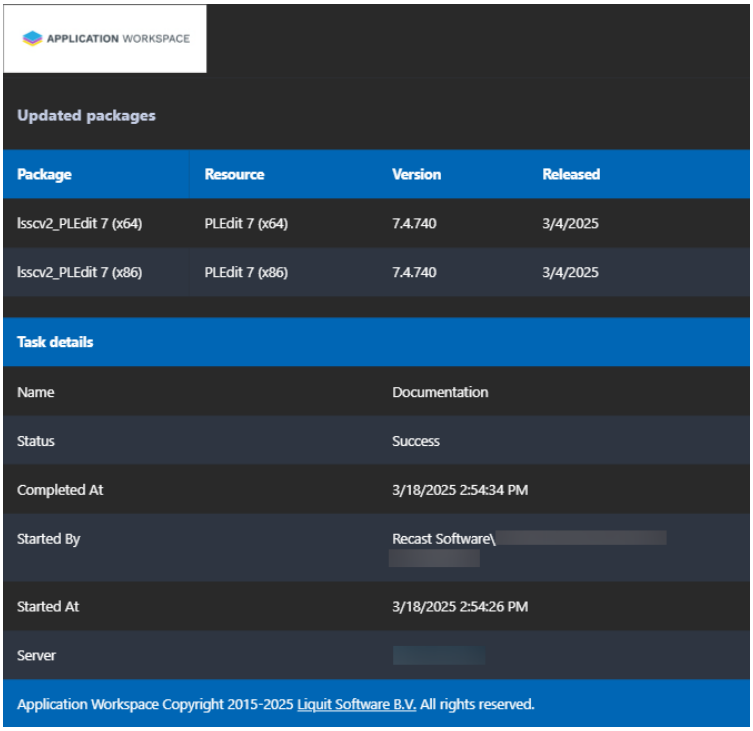
5. In the **Summary**, leave the **Modify package after creation** selected and click **Finish**.

6. Navigate to **Notifications** and enable the notification and add one or more email addresses where Application Workspace should send detailed emails per managed package.

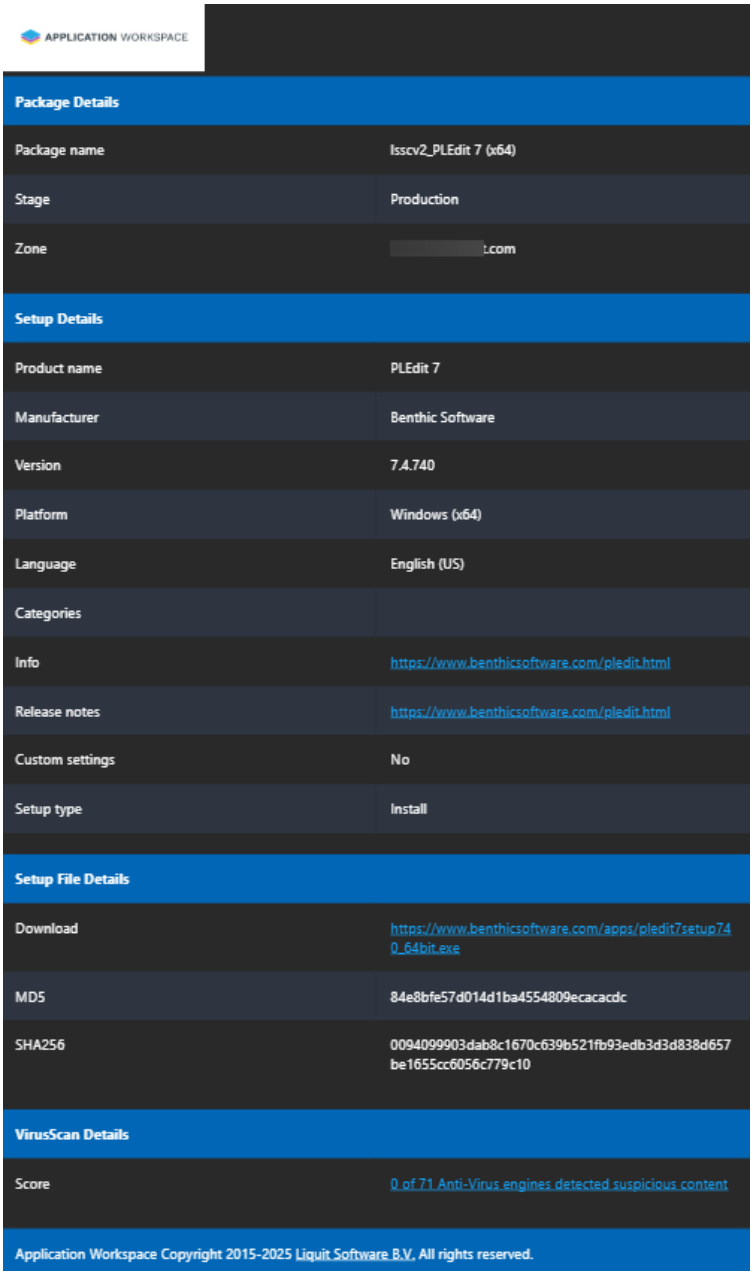
7. Navigate to **Schedule** and add a new schedule where you specify the desired frequency and time.

How the email notification looks like when an automatic update is applied:

Recast


APPLICATION WORKSPACE
Updated packages
Package
Resource
Version
Released
lsscv2_PLEdit 7 (x64)
PLEdit 7 (x64)
7.4.740
3/4/2025
lsscv2_PLEdit 7 (x86)
PLEdit 7 (x86)
7.4.740
3/4/2025
Task details
Name
Documentation
Status
Success
Completed At
3/18/2025 2:54:34 PM
Started By
Recast Software\
Started At
3/18/2025 2:54:26 PM
Server
Application Workspace Copyright 2015-2025 Liquit Software B.V. All rights reserved.

Recast



APPLICATION WORKSPACE

Package Details	
Package name	lsscv2_PLEdit 7 (x64)
Stage	Production
Zone	[redacted]t.com

Setup Details	
Product name	PLEdit 7
Manufacturer	Benthic Software
Version	7.4.740
Platform	Windows (x64)
Language	English (US)
Categories	
Info	https://www.benthicsoftware.com/pledit.html
Release notes	https://www.benthicsoftware.com/pledit.html
Custom settings	No
Setup type	Install

Setup File Details	
Download	https://www.benthicsoftware.com/apps/pledit7setup740_64bit.exe
MD5	84e8bfe57d014d1ba4554809ecacacdc
SHA256	0094099903dab8c1670c639b521fb93edb3d3d838d657be1655cc6056c779c10

VirusScan Details	
Score	0 of 71 Anti-Virus engines detected suspicious content

Application Workspace Copyright 2015-2025 Liquit Software B.V. All rights reserved.

Elevate User Permissions

As an Application Workspace administrator, you can temporarily grant a non-administrator user (meaning a user who is not a member of the local Administrators group) elevated admin privileges that allow them to run specific Application Workspace actions as the current logged-on user. This feature is suited to scenarios where there's a requirement to preserve the logged-on user's identity, such as when installing per-user application or setting up user-context firewall rules, while also allowing administrative operations to run.

> **Security Considerations**
>
> To maintain security while allowing elevated access, this feature includes a number of constraints:

- Once the action completes, the user's elevated permissions do not persist, regardless of whether the action succeeds or fails.
- The feature limits elevated access to the action being run. Privileges do not extend to other processes or sessions and changes made at the Agent Settings level do not allow all existing actions to run with elevated user permissions.
- The user does not become a member of the local Administrators group at any point.
- The elevation mechanism limits execution to approved/allowed programs and restricts token privileges to what is required.

At the same time, administrators must accept inherent risks associated with privilege elevation:

- Actions with scripting, such as script-based installers or a Start Process action with script arguments, can be exploited through malicious scripts executed with elevated rights.
- Open File Explorer extensions may allow a user to launch other applications as an Administrator (for example, opening PowerShell from a context menu with elevated rights).

Enable Process Elevation

An Application Workspace administrator must enable the Process Elevation option, which is disabled and hidden by default.

To enable process elevation:

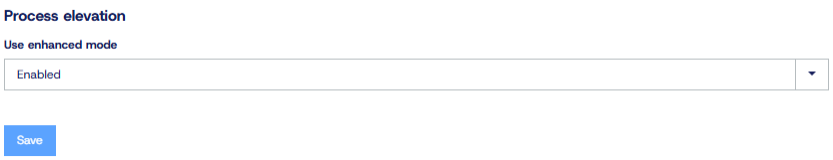
1. Open Application Workspace in its embedded browser via the Smart Icon or the Application Workspace portal.

2. Open the developer console:
   - For an embedded browser (WebWindow) – Click the `F12` key or use the application's developer tools shortcut
   - For a browser-based portal – Click the `F12` key and navigate to the **Console** tab

3. In the browser console, enable Developer Mode by entering the `DEBUG=1` command and clicking the Enter key.

4. In the Application Workspace interface, navigate to **Manage** > **Portal** >**Agent Settings**.

5. Under **Process Elevation**, set **Use enhanced mode** to 'Enabled' and click **Save**. The feature becomes active for the current agent session.



The `DEBUG=1` setting, as well as the feature's visibility in the interface, are limited to a browser session. If the browser is closed or restarted, you'll need to reactivate Developer Mode to access the**User Privilege Elevation** option.

The enabled/disabled state of the feature itself will persist in the Agent configuration.

Configure an Action to Run Within the Elevated User Context

Once you enable process elevation, 'Elevated user' appears as a new Context option in the Action editor.

Supported Actions

You can choose 'Elevated user' as a context for the following actions:

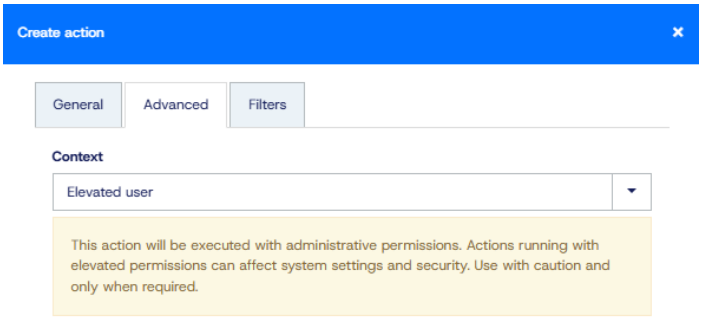
- Install MSI, Install Uploaded MSI, Uninstall MSI
- Install MSP, Install Uploaded MSP
- Install Font, Install Uploaded Font
- Start Process, Start Uploaded Process
- Create Registry Key, Delete Registry Key
- Set Registry Value, Delete Registry Value

To set 'Elevated user' as the context for an action:

1. From the Application Workspace **Packages** page, select a package and choose the **Actions** tab.

2. Open the **Actions** set editor for a supported action type.

3. On the **Advanced** tab, as the Execution or User **Context**, select 'Elevated user' from the drop-down.



NOTE: When the 'Use enhanced mode' option is enabled, selecting 'Elevated User' as the context prompts the Action editor to display an informational message: "This action will be executed with administrative permissions. Actions running with elevated permissions can affect system settings and security. Use with caution and only when required."

4. Click **Confirm** to save the setting.

Notifications for Setup Store Package Updates

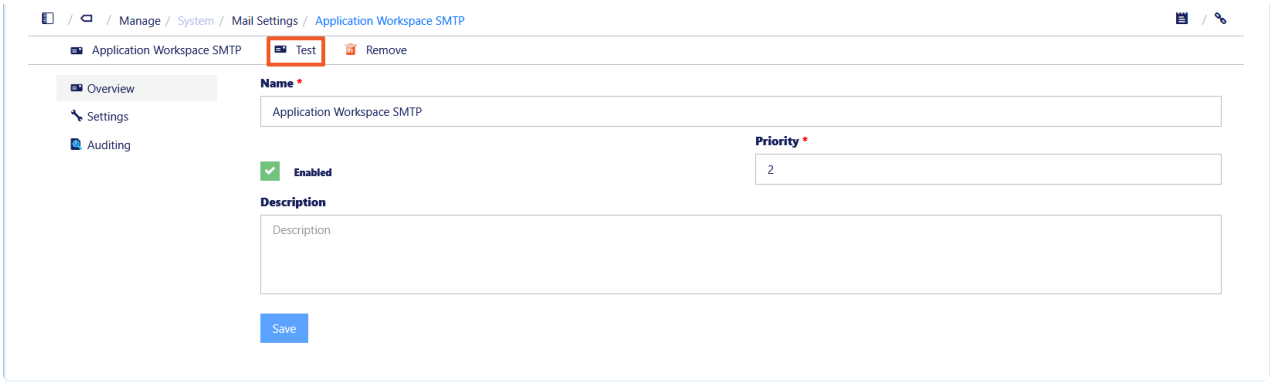
You can create scheduled tasks to receive email notifications when updates are available for the managed application you've created via the Application Workspace Setup Store.

Configure email settings

To receive email notifications from within Application Workspace, you first have to add an email server.

Under **Manage** > **Mail Settings**, click **Create** in the table toolbar and configure all the necessary settings for this email server.
This email server is used by the scheduled tasks to send email messages. For more information, see Mail Settings.

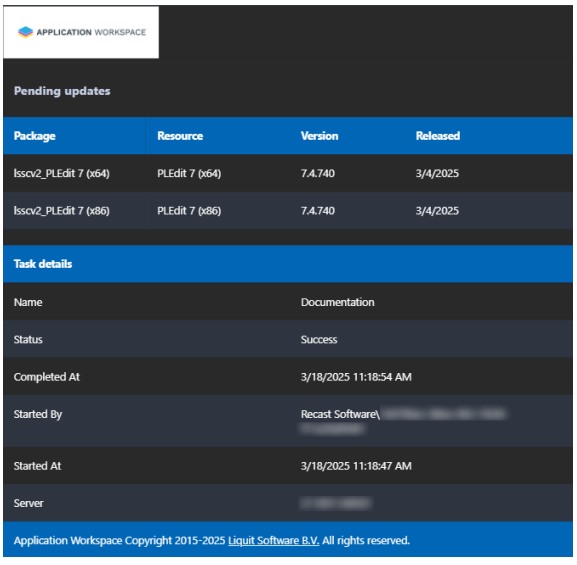
Test the e-mail server
Use the **Test** option on the **Overview** tab to see if the configuration is working correctly.



Create a scheduled task

1. Navigate to **Manage** > **Automation** > **Scheduled Tasks**.
2. Click **Create**.
3. In the **Create scheduled task** that opens, select the **Synchronize connector** type.
4. In **Settings**:
    - Select an existing Application Workspace Setup Store.
    - Select **Check for updates** action.
5. In **Summary**, leave the **Modify package after creation** selected and click **Finish**.
6. Navigate to **Notifications** and enable the notification and add one or more email addresses where Application Workspace should send detailed emails per managed package.
7. Navigate to **Schedule** and add a new schedule where you specify the desired frequency and time.

Example update email notification:



CimFS App Attach package files

Prerequisites

- The certificate with which the MSIX package is signed must be present on the device
- The Package Name and Package family name which you need to create the package for CIM App Attach in Application Workspace. You can use for example MSIX hero.

To create CimFS App Attach package files:

1. Create the package in Application Workspace by navigating to **Manage** > **Packages** and creating a 'Custom' type of package with a descriptive name, like SlimPDFreader (CIM App Attach).

2. Create a new Install-type action set that checks if the MSIX is already installed.

Within the package, create an action set with the following configuration:

- **Overview** tab:
  - **Type**: Install
- **Filters** tab – Create a filter set and add a filter with:
  - **Type**: Windows App Installed
  - **Package Family Name**: SlimPDFreader_1.0.0.0_x64__x1a5e57jmrrvp
  - **Exists**: No

3. Configure an action that creates a temp directory, if one is not present.

Within the Install action set, create an action with:

- **General** tab:
  - **Type**: Create directory
  - **Name**: e.g., Verify Temp Directory
  - **Path**: C:\Temp
- **Advanced** tab:
  - **Context**: Device
- **Filters** tab – Add a filter:
  - **Type**: Directory exists
  - **Path**: C:\Temp
  - **Exists**: Yes
  - **Context**: Device

4. Configure an action that uploads a directory to the Application Workspace Content Store and deploys it to a specified destination on devices.

Within the Install action set, create an action with the following attributes:

- **General** tab:
  - **Type**: Install uploaded directory
  - **Name**: e.g., Install uploaded directory
  - **Path**: ${Documents}\SlimPDFreader
  - **Destination**: C:\Temp\SlimPDFreader

5. Configure an action that mounts a virtual disk to the specified app attach mount point.

Within the Install action set, create an action with:

- **General** tab:
  - **Type**: Attach virtual disk
  - **Virtual disk**: C:\Temp\SlimPDFreader\SlimPDFreader.cim
  - **Mount point**: C:\Temp\appattach

6. Configure an action that stages a Windows app in-place from the mounted path without registering it.

Within the Install action set, create an action with:

- **General** tab:
  - **Type**: Install Windows App

- **Name**: Stage Windows App
- **Path**: C:\Temp\appattach\apps\SlimPDFreader_1.0.0.0_x64__x1a5e57jmrrvp

7. Configure an action that registers the staged Windows app by specifying its Package Family Name.

   Within the Install action set, create an action with:
   - **General** tab:
     - **Type**: Register Windows App
     - **Package Family Name**: SlimPDFreader_1.0.0.0_x64__x1a5e57jmrrvp

8. Create a new Launch-type action set that launches the SlimPDFreader.

   Create an action set with:
   - **Overview** tab:
     - **Type**: Launch

   Within the Launch action set, create an action with:
   - **General** tab:
     - **Type**: Start Windows App
     - **Package Family Name**: SlimPDFreader_1.0.0.0_x64__x1a5e57jmrrvp
     - **Application Id**: SlimPDFreader

9. Move the package from the Development stage to the Production stage and click **Confirm**.

10. Within the package, navigate to **Entitlements** and provide an Active Directory or Entra group.

11. In the **Add entitlement** window, in **Publish**, select where you want to publish your package and click **Confirm**.

Create a category and assign packages to it

Before creating a category, make sure the desired packages are created.

1. Go to **Manage** > **Workspace** > **Categories**.
2. Click **+ Create** in the table toolbar.
3. In the **Create category** dialog box that opens, in **Overview**, fill in the necessary info. Click **Next**.
4. In **Summary**, leave the **Modify category after creation** selected. Click **Next**.
5. In the detailed view screen that opens, in **Packages**, search for the necessary package and add it.
6. Double-click the entry of the package to open it.
7. Go to **Entitlements** and add the necessary identities. This step makes the category available for the user in the Side Menu of **Workspace** or **Catalog**, depending on where you configured the package to be published.

Create a license and assign packages to it

Before creating a license, make sure the desired packages are created.

1. Go to **Manage** > **Workspace** > **Licenses**.
2. Click ✚ **Create** in the table toolbar.
3. In the **Create license** dialog box that opens, in **Overview**, fill in the necessary info. Click **Next**.
4. In **License** fill in the necessary info and click **Next**.
5. In **Summary**, leave the **Modify license after creation** selected. Click **Next**.
6. In the detailed view screen that opens, in **Overview**, insert the license key.
7. In **Packages**, search for the necessary package and add it.
8. Double-click the entry of the package to open it.
9. Go to **Entitlements** and add the necessary identities and select the Catalog publish stage for them. This step makes the category available for the user in the Side Menu of **Catalog**.

Add an App-V Package

Prerequisite

- Microsoft Application Virtualization Sequencer must be installed.

Obtain Information from an App-V Package Using Microsoft Application Virtualization Sequencer

1. Open **Microsoft Application Virtualization Sequencer**.
2. Select **Modify an Existing Virtual Application Package**.
3. Click **Edit Package**, then click **Next**.
4. Click **Browse** and locate the App-V package. and double-click on it.
5. Click **Edit**.
6. From the Properties tab, save the **Package Name** and **Package GUID** as you will need them in the later steps.
7. Navigate to the **Package Files** tab and expand the folders until you locate the executable.
8. Right-click the executable (for example, `XenCenterMain.exe`) and click **Edit**.
9. Save the path as you will need it later. Prefix the path with `Root\VFS\` and remove `{}` from `ProgramFilesX86`

Example: `Root\VFS\ProgramFilesX86\XCP-ng Center\XenCenterMain.exe`

Add an App-V Package in Application Workspace

1. Open Application Workspace and navigate to **Manage** > **Packages**.

2. Create a **Custom** package with the following action sets and actions:
    - Action set type: Install
        - Action type: **Install uploaded file** with **Destination** set to `${PackageTempDir}\`
        - Action type: **Run uploaded script** with the following properties:
            - **Script engine** set to **Windows PowerShell**
            - **Script:** `EnableAppv`
            - **Context**: Device
        - Action type: **Run uploaded script** with the following properties:
            - **Script engine** set to **Windows PowerShell**
            - **Script:** **`AddAppvClientPackage "XCP-ng Center.appv"`** where you replace the App-V Package name with your own package name
            - **Directory** set to `${PackageTempDir}\`
            - **Context**: Device
        - Action type: **Run uploaded script** with the following properties:
            - **Script engine** set to **Windows PowerShell**
            - **Script:** **`PublishAppvClientPackage -Name "XCP-ng Center" -Global`** where you replace the App-V Package name with your own package name
            - **Context**: Device
    - Action set type: Launch (This step uses the **Package GUID** and **Package Path** collected in the previous section)
        - Action type: **Start process** with the following properties:
            - **Process**: C:\ProgramData\Microsoft\AppV\Client\Integration\<Package GUID>\<Package Path>
            - In **Filters** tab create a *File exist* type filter where you specify the path to the executable

Example:

```
C:\ProgramData\Microsoft\AppV\Client\Integration\
728497f7-c17f4b3c-bef4-c572abd4dd9a\
Root\VFS\ProgramFilesX86\XCP-ng Center\XenCenterMain.exe
```

Uninstall action set (Optional)

Within the same Custom package, add the following action set and actions:

- Action set type: Uninstall
    - Action type: **Run uploaded script** with the following properties:

- **Script engine** set to **Windows PowerShell**
- **Script:** `Get-AppvClientPackage -name "XCP-ng Center" | Stop-AppvClientPackage | Remove-AppvClientPackage` where you replace the App-V Package name with your own package name
- **Context**: Device

Deploy a package with hidden entitlement

Scenario

You have a script which synchronizes to OneDrive the SharePoint sites the user can access. The script also removes the SharePoint sites the user is not entitled to anymore, but which are still synced to OneDrive.

You need this script to run automatically on the user devices without relying on user interaction.

Steps

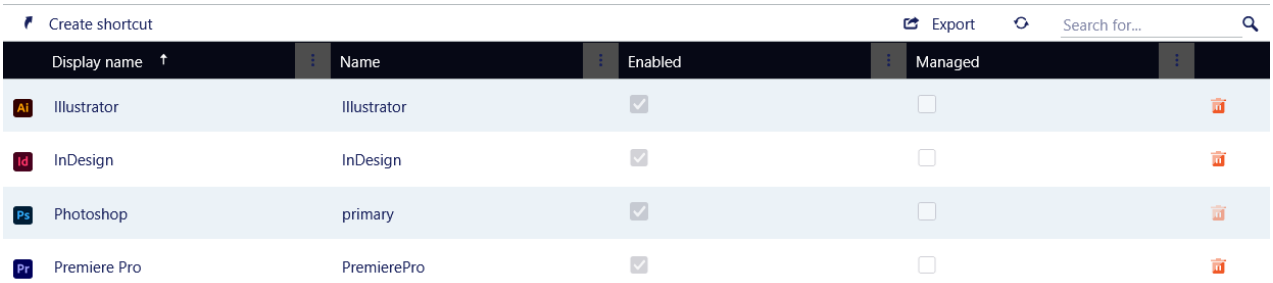
1. Add a new entitlement to the package.
2. In the **Add entitlement** dialog box that opens,
    - In the **Overview** tab, **Publish** field, select *Hidden*.
    - In the **Events** tab add a 'Application Workspace Logon' type event, with a 'Launch' type action.
3. Click **Confirm**.

The package will start when the entitled user logs in to Application Workspace, without requiring manual user interaction. Now the package will not be visible in the Application Workspace, since no user interaction is needed.

Deploy a package with multiple shortcuts

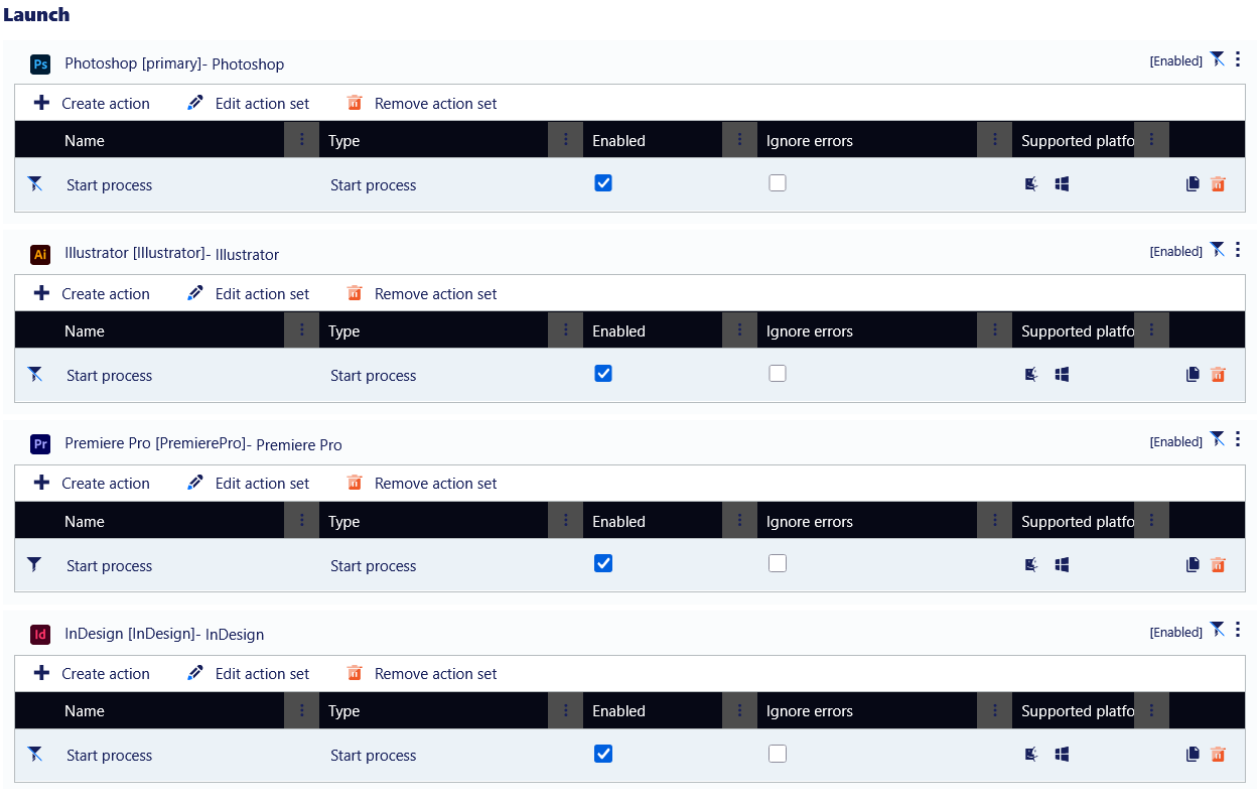
You can deploy multiple shortcuts for a single package. Let's say you create an Adobe Creativity and Design suite package and you want to make several applications available to your users as individual smart icons: Photoshop, Premiere Pro, Illustrator, InDesign.

1. Go to **Manage** > **Workspace** > **Packages**.
2. Create a custom package called *Adobe Creativity and Design suite* and select an icon for it.
3. Go to **Entitlements** and assign your desired identities.
4. Go to **Release** > **Shortcuts**.
5. Open the primary shortcut and name it *Photoshop* and select an icon for it. Create additional shortcuts for Premiere Pro, Illustrator, InDesign and set icons for them.



6. Go to **Releases** > **Actions**.

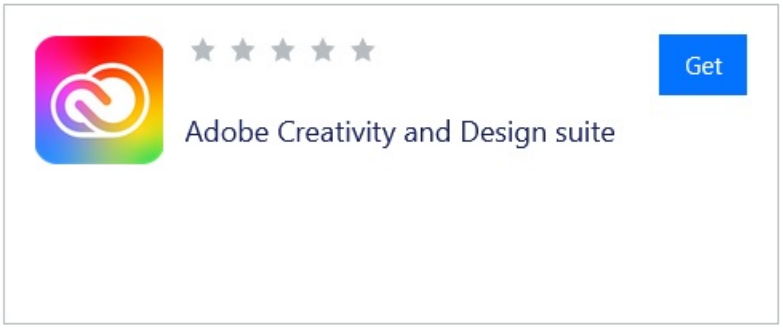
7. Create 4 *Launch* type action sets for each application and select the corresponding shortcuts created at steps 4–5.



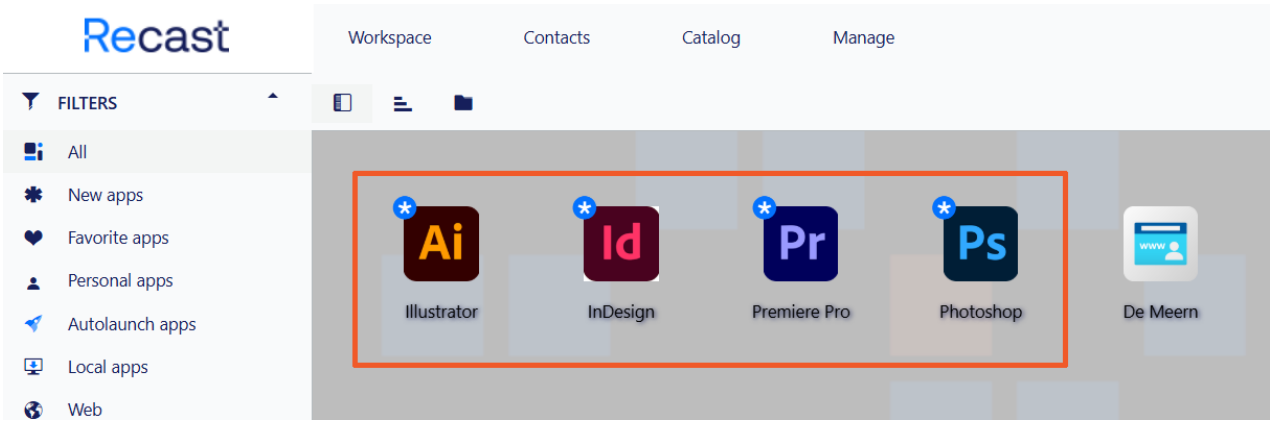
8. Inside each action set create a *Start process* action for all the 4 applications.

9. Move the *Development* stage to *Production*, to make the package available to your users.

On the user's device, in the **Catalog** tab of Application Workspace, the Adobe Creativity and Design suite package will be displayed as a single item.



After the user gets the package from the **Catalog** tab to the **Workspace** tab, 4 new smart icons will be displayed in the **Workspace** tab and each one will open a different app.



Shortcuts can be set only in *Launch* type actions/action sets.

Deploy Custom Fonts

There are multiple ways in which you can deploy fonts across devices. For example, you can use theRegisterFonts action inside an MSI. This is the official Microsoft-supported way of installing a particular font, which not only copies the file to the `C:Windows\Fonts` folder but also takes care of the registration process.

But using the RegisterFonts action would mean creating an MSI package with one of your favourite MSI packaging tools. An alternative is to use PowerShell.

Within Application Workspace, the process is simpler.

In an install type action set you can use theInstall font or Install uploaded font.

- Install font can be used when you want to fetch the .TTF file of the selected font from a local drive.
- Install uploaded font can be used to upload the font to the Content Store and install it on devices from the content repository.

Upload config.xml in Setup Store App

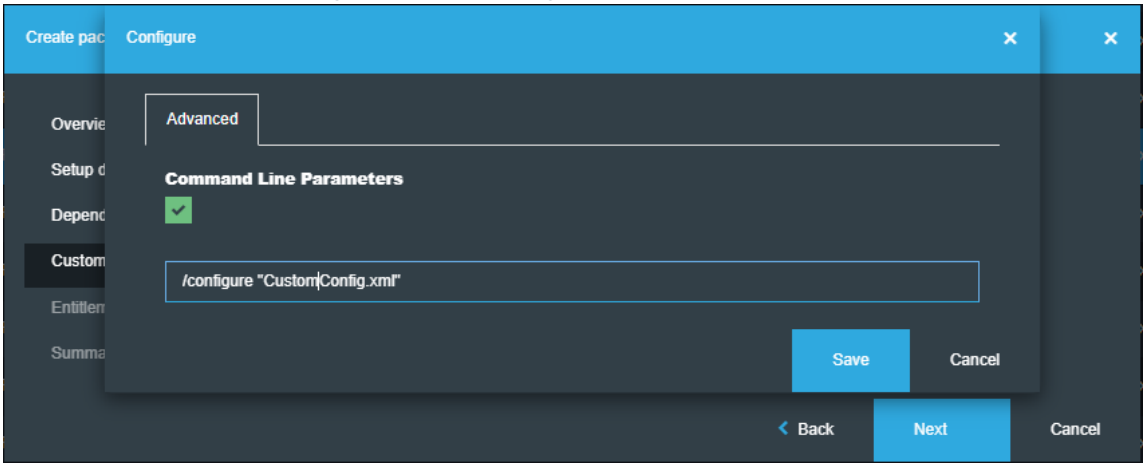
Scenario: within a Microsoft Office managed package you need a pre-install action which installs a custom config.xml on the user device. Then you can edit the install string of the clicktorun.exe process to point to that xml instead.

Currently, the only way to include your own custom configuration .xml file within a managed package is to add an extra action set that copies
the file you created with the ODT webtool at Microsoft 365 Apps admin center, to the the desired location, for example ${PackageTempDir}.

Steps

1. Open the Application Workspace Setup Store connector and navigate to **Overview** > **Resources**.
2. Select a Microsoft 365 Apps product and click **Create package**.
3. In the **Create package** dialog box that opens
    - in the **Customize** screen, modify the command line parameters to `/configure "CustomConfig.xml"` where you replace CustomConfig.xml with the name of your file. Note that when no variable is used, it uses the working

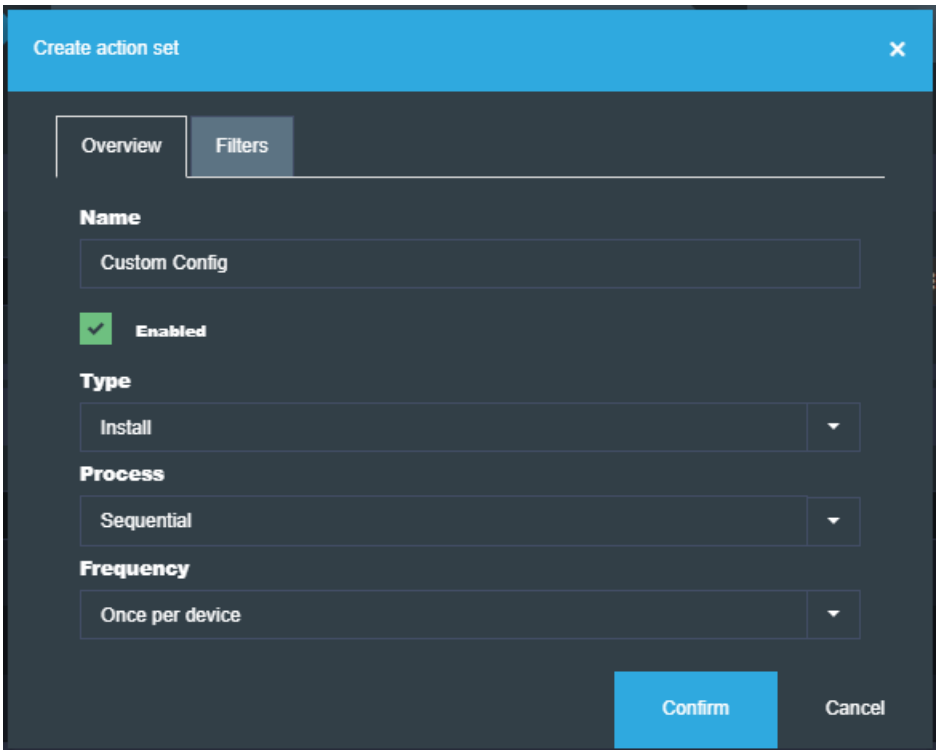
directory specified in the package, which is ${PackageTempDir}



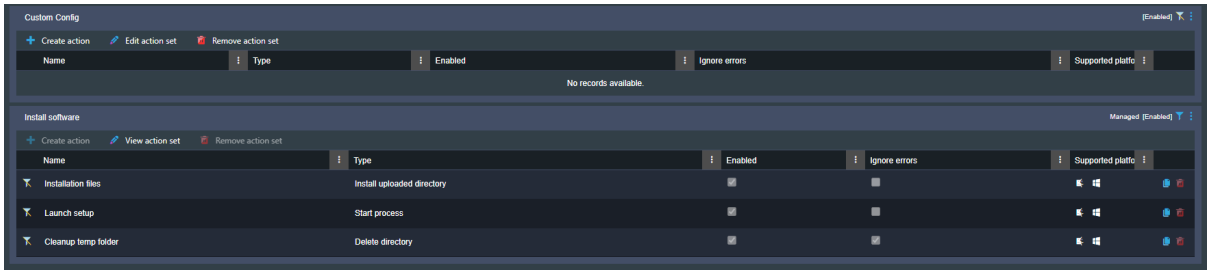
> In the **Summary** screen, leave the "Modify package after creation" option selected, so that the newly created package opens automatically after you click **Finish**.

4. Within the package, navigate to **Releases** > **Actions**.

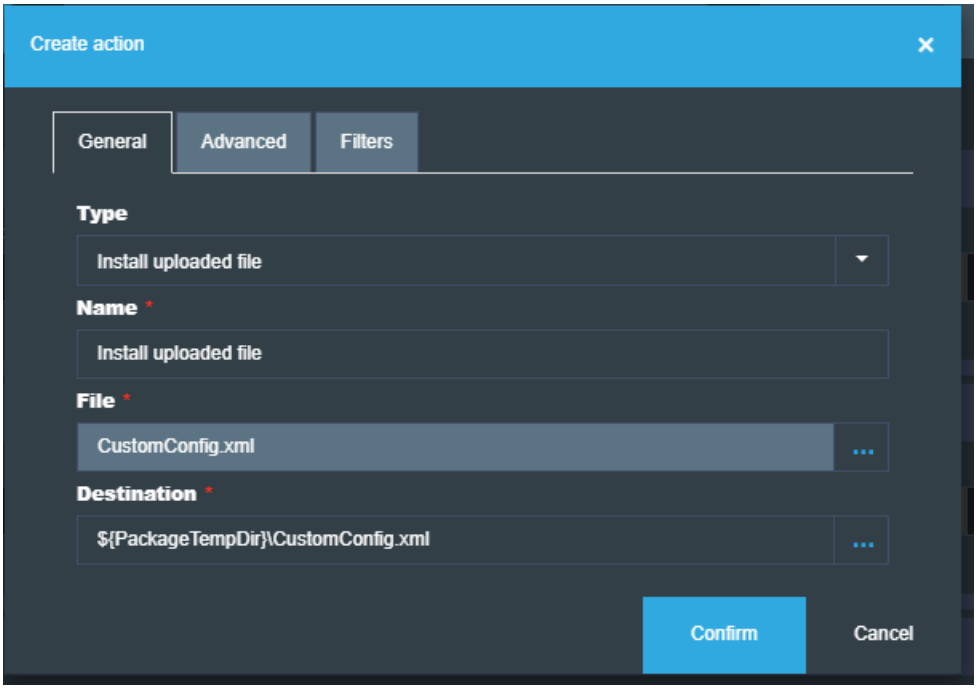
5. Create a new Install type action set named Custom Config.



6. Move the *Custom Config* action set above the *Install software* action set, by clicking on the title bar of the action set and dragging and dropping it in the new location.

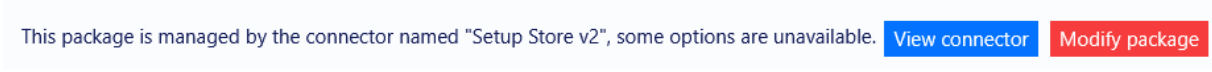


7. In the table toolbar of the action set, click **+ Create action**, and an *Install uploaded file* type of action. Browse for the file you want to upload and specify the destination path where you want the file to be placed.



Note that in the case of managed packages, existing action sets cannot be edited or deleted. You can still add new action sets to the package.

To convert a package from managed to unmanaged you must by click on **Modify package** at the top of the **Overview** screen.

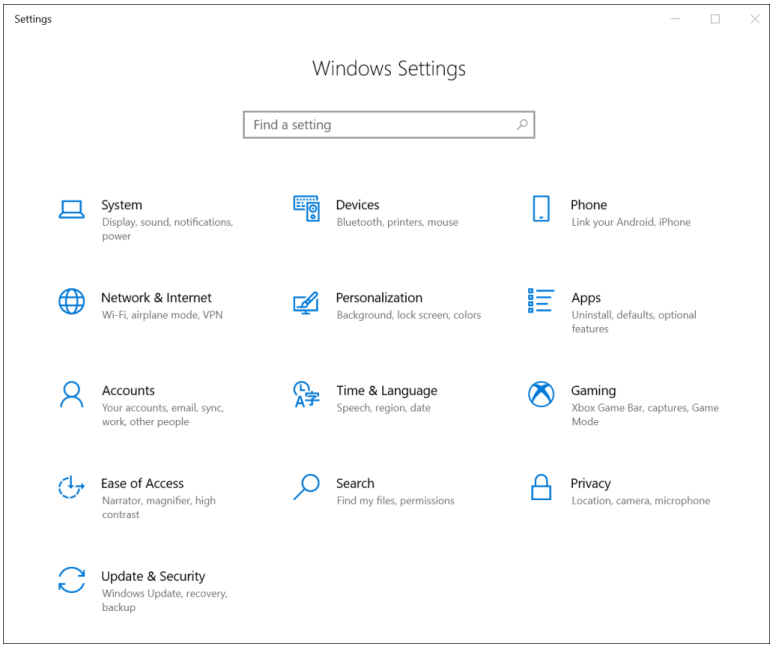


By becoming unmanaged, future Application Workspace Setup Store updates will not be applied to the app.

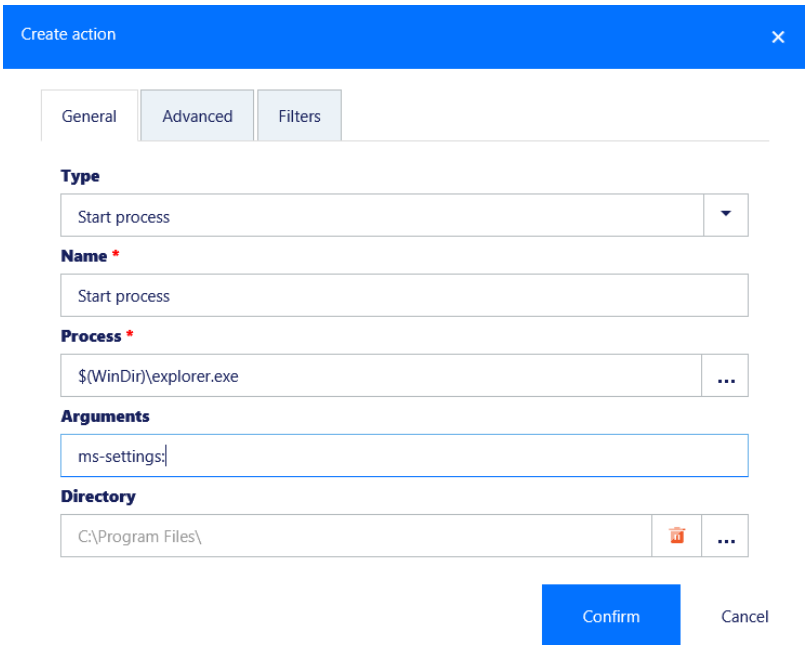
Launch Windows Settings

You can start any of the Windows Settings UWP apps using explorer.exe and an URI parameter.

Recast



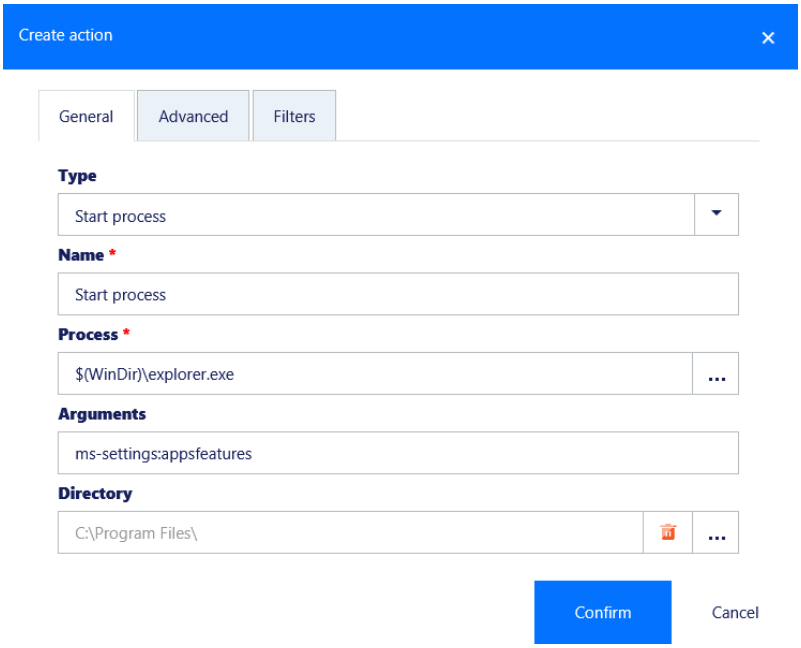
In this article we give you some examples of such settings which are started using `ms-settings:`



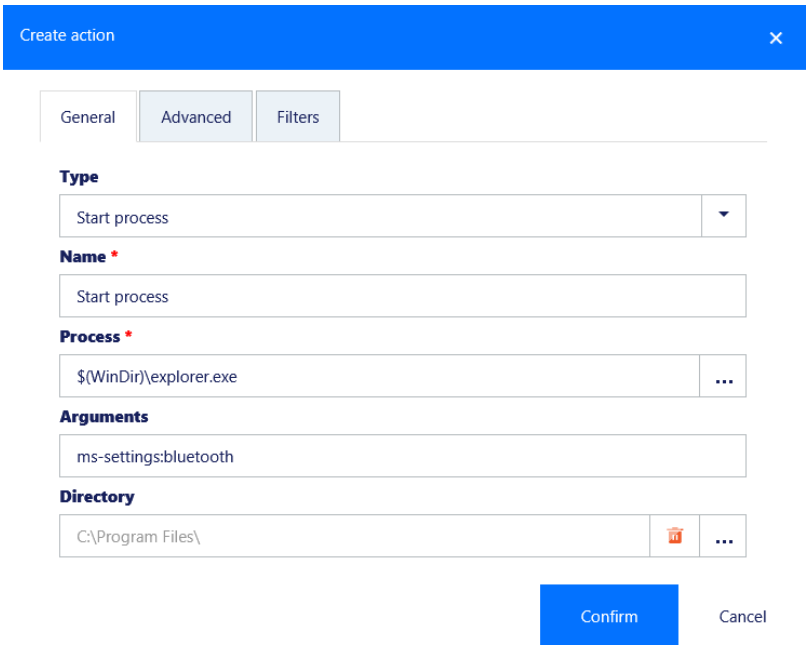
Examples

Apps & Features Settings can be started using `ms-settings:appsfeatures`

Recast



Bluetooth Settings can be started using `ms-settings:bluetooth`



See also

Complete list of ms-settings URIs used to open various pages of the Settings app inLaunch Windows Settings - Windows apps | Microsoft Learn.