

# Uninstall add-ins under development

Article • 05/19/2025

Incompletely removed add-ins can leave artifacts on your computer, such as custom ribbon buttons or registry entries, during development. In this article, we call these "ghost add-ins".

Outlook add-ins also might add these artifacts to other computers when you sign into Outlook on them with the same ID as you used to develop the add-in.

## Important

When you sign into Outlook, it downloads from Exchange, and sideloads, all the Outlook add-in manifests that are associated with your ID, *including add-ins that you are developing on a different computer using the same ID*. For example, any custom ribbon buttons defined in the manifest will appear for the add-in.

If the URLs in the manifest point to a non-localhost server and that server is running and accessible to the non-development computer, then Outlook caches the add-in's files in the local file system and the add-in usually runs normally on the computer. Otherwise, the add-in doesn't function, but visible parts of it, such as custom ribbon buttons appear. They have the labels defined in the manifest. The add-in's button icons also appear if they were ever cached locally on the non-development computer and the cache was never cleared. Icon files aren't stored with Exchange, so if they were never cached on the non-development computer (or the cache has been cleared), then the buttons have default icons.

Until the add-in's registration is removed from Exchange, the add-in will continue to appear. See [Remove a ghost add-in](#) for information about removing the registration in Exchange.

This article provides some guidance to minimize the chance of these problems and to resolve them if they occur.

## Prevent the problems

When an add-in is sideloaded, several things happen:

- A web server, usually on localhost, is started to serve the add-in's files (such as the HTML, CSS, and JavaScript files).
- These same files are cached on your development computer.

- The add-in is registered with the development computer. The registration is done with Registry entries on a Windows computer or with certain files saved to the file system on a Mac.
- Most tools for sideloading add-ins automatically open the Office application that the add-in targets. The tools also populate the application with any custom ribbon buttons or context menu items that are defined in the add-in's manifest.
- For an Outlook add-in, the add-in's manifest is registered with the Exchange service.

## Use your tool's uninstall facility

To prevent ghost add-ins, end every testing, debugging, and sideloading session by using the uninstall (also called unacquire) option that is provided by the tool that you used to start the session. Doing this reverses the effects of sideloading, as stated earlier in this article.

The following list identifies, for each tool, how to uninstall but doesn't describe the procedures or syntax in detail. *Be sure to use the links to get complete instructions.*

### ⓘ Note

Some of these tools don't close the Office application that opened automatically. In that case, close the application manually immediately after ending the session.

- **Yeoman generator for Office Add-ins (Yo Office):** Use the `npm stop` script at the same command line where you started the session with `npm start`. For more information, see the various articles in the **Get started** and **Quick starts** sections and [Remove a sideloaded add-in](#).
- **Microsoft 365 Agents Toolkit for Visual Studio Code:** Select **Run | Stop Debugging** in Visual Studio Code. For more information, see the last step of [Create an Outlook Add-in project](#) which also applies to non-Outlook add-ins.
- **Office Add-in Development Kit for Visual Studio Code:** With the Office Add-in Development Kit extension open, select **Stop Previewing Your Office Add-in**. For more information, see [Stop testing your add-in](#).
- **office-addin-debugging tool:** Use the `office-addin-debugging stop` command at the same command line where you started the session with `office-addin-debugging start`. For more information, see [Sideload with the Office-Addin-Debugging tool](#).
- **Microsoft 365 Agents Toolkit CLI:** Use the `atk uninstall` command at the same command line where you started the session with `atk install`. For more information, see [Sideload with Microsoft 365 Agents Toolkit CLI](#).
- **Visual Studio:** Select **DEBUG | Stop debugging** in the menu, or press `Shift + F5`, or click the square red "stop" button on the debugging bar. Alternatively, closing the Office

application also stops the session and uninstalls the add-in. For more information, see [First look at the Visual Studio debugger](#).

## Remove a ghost add-in

To remove a ghost add-in, you need to remove the artifacts that were created when it was last sideloaded, remove its local registration, and for Outlook add-ins remove its registration in Exchange.

### Tip

There's a fast way to remove a ghost add-in on Windows computers if the add-in was installed with the Teams Toolkit CLI. Try this first, and if it works, you can skip the remainder of this section.

1. Obtain the add-in's title ID from the Registry key

**HKEY\_CURRENT\_USER\SOFTWARE\Microsoft\Office\16.0\Wef\Developer\Outlook SideloadManifestPath\TitleId**. (The string "Outlook" is in the key name for historical reasons, but it applies to any add-in installed with the Agents Toolkit CLI.)

2. Run the following command in a command prompt, bash shell, or terminal. Replace "{title ID}" with the title ID of the add-in including the "U\_" prefix; for example,

`U_90d141c6-cf4f-40ee-b714-9df9ea593f39`.

command line

```
atk uninstall --mode title-id --title-id {title ID} --interactive false
```

The process for removing the add-in varies depending on whether the add-in is for Outlook or some other Office application.

### Note

In the [unified manifest for Microsoft 365](#), an add-in can be configured to support Outlook and one or more other Office applications; that is, there's more than one member of the `"extensions.requirements.scopes"` array in the manifest and one of the members is `"mail"` (or the `"extensions.requirements.scopes"` property isn't present). Treat an add-in that is configured in this way as an Outlook add-in.

If the ghost add-in is not an Outlook add-in, skip to the section [Remove the add-in artifacts](#).

## Remove the Exchange registration of a ghost Outlook add-in

1. Log into Outlook with the same ID you used when you sideloaded the add-in.
2. Open PowerShell as an Administrator.
3. Run the following commands. Answer "Yes" to all confirmation prompts.

PowerShell

```
Install-Module -Name ExchangeOnlineManagement -RequiredVersion 3.4.0
Set-ExecutionPolicy RemoteSigned
Connect-ExchangeOnline
```

### ⓘ Note

If the `Connect-ExchangeOnline` command returns the error "ActiveX control '8856f961-340a-11d0-a96b-00c04fd705a2' cannot be instantiated because the current thread is not in a single-threaded apartment", just run the command a second time. This is a well-known bug.

4. Run the following command. Answer "Yes" to all confirmation prompts.

PowerShell

```
Get-App | Format-Table -Auto DisplayName,AppId
```

A list of the add-ins installed on Outlook displays. These will include built-in Microsoft add-ins and add-ins you have installed. Any ghost Outlook add-ins will also be listed.

5. Find the ghost add-in in the list. If it was created with Yo Office or another Microsoft tool, it probably has the name "Contoso Task Pane Add-in".
6. Copy the App ID (a GUID) of the add-in. You need it for later steps.
7. Run the command `Remove-App -Identity {{The GUID OF YOUR ADD-IN HERE}}` (e.g., `Remove-App -Identity 26ead0cb-10dd-4ba2-86c6-4db111876652`). This command removes the add-in from Exchange.

 **Warning**

The removal of the registration needs to propagate to all Exchange servers. Wait at least three hours before continuing with the next step.

8. Continue with the section [Remove the add-in artifacts](#).

## Remove the add-in artifacts

### Important

Carry out this procedure on all devices on which you have had the add-in sideloaded.

1. Log out from all Office applications and then close them all, including Outlook.
2. [Clear the Office cache](#). If the ghost add-in supports Outlook, use [Clear the cache in Outlook manually](#).
3. Continue with the section [Remove the local registration](#).

## Remove the local registration

### Important

Carry out this procedure on all computers on which you have had the add-in sideloaded.

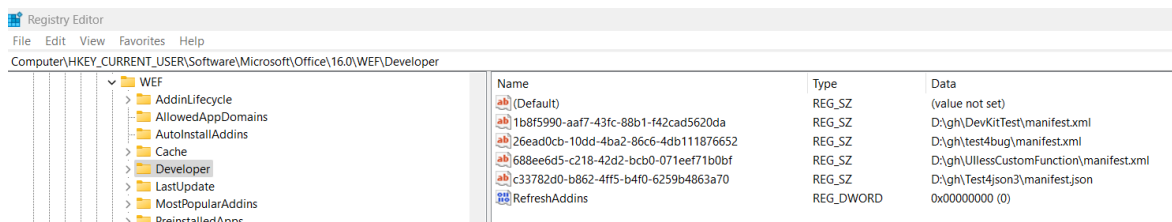
1. Delete the local registration of the ghost add-in. The process varies depending on the operating system.

#### Windows

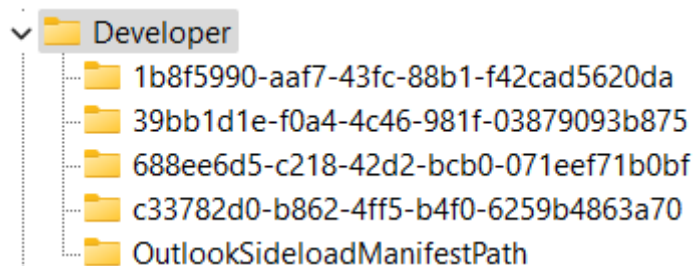
- a. Open the **Registry Editor**.
- b. Navigate to **Computer\HKEY\_CURRENT\_USER\Software\Microsoft\Office\16.0\WEF\Developer**. This key lists the add-ins that are currently sideloaded, or were sideloaded in the past and weren't fully uninstalled. The **Data** value for each entry is the path to the add-in's manifest. The **Name** value varies depending on which version of which tool was used to create and sideload the add-in. If Visual Studio was used, the name is typically is also the path to the manifest. For other tools, the name is typically the add-in's ID. When an Office application launches, it reloads all add-ins listed in this key (that support the Office application). Reloading may have no practical or discernable effect if the add-in's artifacts have been deleted from the

cache, or the manifest no longer exists at the path, or the add-in's files aren't being served by a server.

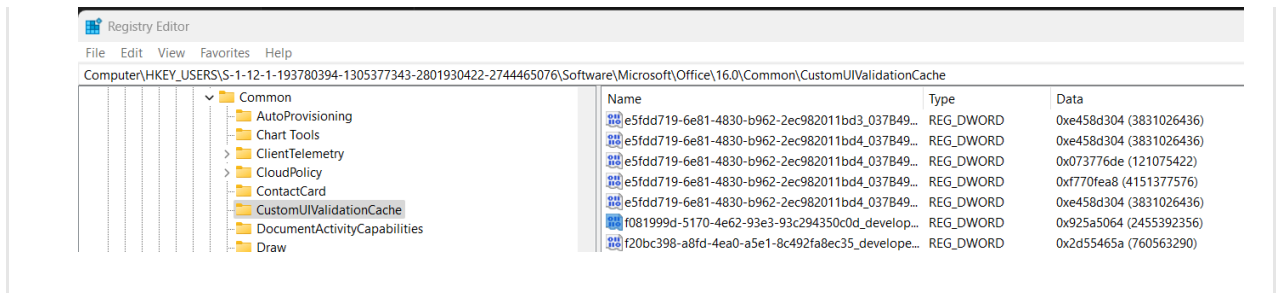
Find the entry for the ghost add-in and delete it. If it is an Outlook add-in, then you have the ID from [removing the Exchange registration](#). You can also use the path in the **Data** column to find the manifest to help identify the add-in the entry refers to and read the ID from the manifest. If any manifests listed in the **Data** column no longer exist at the specified path, then delete the entries for those manifests.



- c. Expand the ... **Developer** node in the registry tree. Look for a subkey whose name is the same ghost add-in's ID. If it is there, delete it.



- d. Navigate to **Computer\HKEY\_USERS\{SID}\Software\Microsoft\Office\16.0\WEF\Developer**, where {SID} is the [SID](#) of the user you were signed in with when you sideloaded the add-in, and repeat the preceding two steps.
- e. Navigate to **Computer\HKEY\_CURRENT\_USER\Software\Microsoft\Office\16.0\Common\CustomUIValidationCache**. In the **Name** column, find all the entries that begin with the add-in's ID (a GUID) and delete them. Then navigate to **Computer\HKEY\_USERS\{SID}\Software\Microsoft\Office\16.0\Common\CustomUIValidationCache**, where {SID} is the SID of the user you were signed in with when you sideloaded the add-in, and repeat the process.



2. If you are removing an Outlook add-in, continue with the section [Test for removal of Outlook add-ins](#).

## Test for removal of Outlook add-ins

Open Outlook with the same identity you used when you created the add-in. If artifacts from the add-in (such as custom ribbon buttons) reappear after a few minutes or if event handlers from the add-in seem to be active, then the removal of the add-in's registration from Exchange hasn't propagated to all Exchange servers. Wait at least three hours and then repeat the procedures in the sections [Remove the add-in artifacts](#) and [Remove the local registration](#) on the computer where you observed the artifacts.

## See also

- [Troubleshoot development errors with Office Add-ins](#)
- [Clear the Office cache](#)
- The PowerShell reference for [Install-Module](#), [Set-ExecutionPolicy](#), [Connect-ExchangeOnline](#), and [Get-App](#).