# Test and debug Office Add-ins on a non-local server

Article • 05/19/2025

When you've completed development and testing on a localhost and want to stage and test the add-in from a non-local server or cloud account, you can use the tool office-addin-debugging ⧉ for any Node.js-based add-in project. (The tool isn't supported in projects created with Visual Studio.)

> ⓘ **Note**
>
> If you're working on a Windows computer, you may have another option for non-local testing. See **Sideload Office Add-ins for testing from a network share**.

## Projects created with Microsoft 365 Agents Toolkit or the Office Yeoman Generator (Yo Office)

If your project was created with Agents Toolkit or Office Yeoman Generator (Yo Office), then the office-addin-debugging tool is already installed and your package.json file has `start` and `stop` scripts that invoke the tool. To use it for non-local testing, update the domain part of the URLs in your manifest to point to your staging server (or CDN as needed). Then run `npm run start` at the command line (or Visual Studio Code TERMINAL) to sideload the add-in for testing and debugging.

> ⓘ **Important**
>
> The office-addin-debugging tool registers the add-in in the Windows registry or a special folder on a Mac. For an Outlook add-in, it also registers the add-in in Exchange. To avoid subtle bugs when developing, always end a testing session by running `npm run stop` to ensure that these registrations are removed and that the server process is fully stopped. *Manually closing the server, the command line window (or TERMINAL), Visual Studio Code, or the Office application doesn't remove these registrations.*

## Other projects

If your project wasn't created with Agents Toolkit or Yo Office, run the tool with npx in the root of the project. Invoke it with its `start` command followed by the relative path to the manifest.

The following is an example.

```command line
npx office-addin-debugging start manifest.json
```

This command sideloads the add-in for testing and debugging. The tool also works with an add-in only manifest.

There are many options for the `start` command. For details, see the README for the tool at office-addin-debugging ⧉ .

> ⓘ **Important**
>
> The office-addin-debugging tool registers the add-in in the Windows registry or a special folder on a Mac. For an Outlook add-in, it also registers the add-in in Exchange. To avoid subtle bugs when developing, always end a testing session by running `npx office-addin-debugging stop` to ensure that these registrations are removed and that the server process is fully stopped. *Manually closing the server, the command line window (or TERMINAL), Visual Studio Code, or the Office application doesn't remove these registrations.* If you used the `--prod` option with the `start` command, use the same option with the `stop` command.

# Clear the Office cache

06/25/2025

The Office cache stores resources and data used by Office Add-ins. By accessing stored resources, an add-in's performance is improved as it avoids redownloading these resources when needed.

You should clear the Office cache in the following scenarios.

- When you want to remove an add-in that you've previously sideloaded on Windows, Mac, or iOS.
- When you've updated the manifest (for example, to update the file names of icons or text of add-in commands). This ensures that you're using the latest version of the add-in.

> 💡 **Tip**
>
> For add-ins that implement a task pane, if you only want the sideloaded add-in to reflect recent changes to its HTML or JavaScript source files, you shouldn't need to clear the cache. Instead, put focus in the add-in's task pane (by selecting anywhere within the task pane). Then, select `Ctrl` + `F5` to reload the add-in.

- When you want to resolve issues or errors when running the add-in.

> ⓘ **Note**
>
> To remove a sideloaded add-in from Excel, OneNote, PowerPoint, or Word on the web, see **Sideload Office Add-ins in Office on the web for testing: Remove a sideloaded add-in**.
>
> To remove a sideloaded add-in from Outlook on the web, see **Sideload Outlook add-ins for testing**.

> ⊗ **Caution**
>
> When you clear the Office cache, clear it completely. Don't delete individual manifest files. This can cause all add-ins to stop loading.

## Types of caches

The Office cache can refer to either the web cache or the Wef cache.

- The **web cache** temporarily stores web-based resources and data used by an individual Office Add-in.
- The **Wef cache** locally stores resources and data for all installed Office Add-ins.

The following table outlines which Office cache types can be cleared on different platforms. It also provides links to instructions on how to clear a specific cache.

⌞⌝ Expand table

| Platform | Types of caches to clear | Options to clear the cache |
|---|---|---|
| Windows | Both the web and Wef caches. There's currently no option to clear one cache without clearing the other. | <ul><li>Automatically clear the cache</li><li>Manually clear the cache</li><li>Use the Microsoft Edge developer tools on Windows 10</li></ul> |
| Mac | <ul><li>Web</li><li>Both web and Wef caches</li></ul> | <ul><li>**Web**: Use the personality menu to clear the web cache</li><li>**Web and Wef**: Clear the web and Wef caches on Mac</li></ul> |
| iOS | <ul><li>Web</li></ul> | <ul><li>Use JavaScript to clear the cache on iOS</li></ul> |

# Clear the Office cache on Windows

Depending on your Office host and operating system, you can automatically or manually clear both the web and Wef caches on a Windows computer.

> ⓘ **Important**
>
> On Windows, the automatic and manual options clear both the web and Wef caches. There's currently no option to clear one cache without clearing the other.

## Automatically clear the cache

> ⊘ **Note**
>
> The automatic option is only supported for Excel, PowerPoint, and Word. Outlook only supports the **manual option**.

This method is recommended for add-in development computers. If your Office on Windows version is 2108 or later, the following steps configure the Office cache to be cleared the next time Office is reopened.

1. From the ribbon of Excel, PowerPoint, or Word, navigate to **File** > **Options** > **Trust Center** > **Trust Center Settings** > **Trusted Add-in Catalogs**.
2. Select the **Next time Office starts, clear all previously-started web add-ins cache** checkbox.
3. Select **OK**.
4. Restart Excel, PowerPoint, or Word.

## Manually clear the cache

### Manually clear the cache in Excel, Word, and PowerPoint

To remove all sideloaded add-ins from Excel, Word, and PowerPoint, delete the contents of the following folder.

```
%LOCALAPPDATA%\Microsoft\Office\16.0\Wef\
```

If the following folder exists, delete its contents, too.

```
%userprofile%\AppData\Local\Packages\Microsoft.Win32WebViewHost_cw5n1h2txyewy\AC\#
!123\INetCache\
```

### Manually clear the cache in Outlook

Before attempting to clear the cache in Outlook, first try to remove the sideloaded add-in using the steps outlined in Sideload Outlook add-ins for testing.

If this add-in removal doesn't work, then delete the contents of the `Wef` folder as noted for Excel, Word, and PowerPoint in Manually clear the cache in Excel, Word, and PowerPoint.

If your Outlook add-in uses the Unified manifest for Microsoft 365, also delete the following folder.

```
%userprofile%\AppData\Local\Microsoft\Outlook\HubAppFileCache
```

To clear the cache in [new Outlook on Windows ↗](#), perform the following steps.

1. Close the Outlook client if it's open.

2. From a command line, run the following:

   ```command line
   olk.exe --devtools
   ```

   This opens the new Outlook on Windows client and an instance of the Microsoft Edge DevTools.

3. In the Microsoft Edge DevTools window, select the **Network** tab.

4. Select and hold (or right-click) anywhere in the **Requests** table. Then, select **Clear browser cache**.

## Use the Microsoft Edge developer tools on Windows 10

To clear the Office cache on Windows 10 when the add-in is running in Microsoft Edge, use the Microsoft Edge DevTools.

> ⓘ **Note**
>
> To clear the Office cache using the following steps, your add-in must have a task pane. If your add-in is a UI-less add-in -- for example, one that uses the **on-send** feature -- you'll need to add a task pane to your add-in that uses the same domain for **SourceLocation**, before you can use the following steps to clear the cache.

1. Install the [Microsoft Edge DevTools ↗](#).

2. Open your add-in in the Office client.

3. Run the Microsoft Edge DevTools.

4. In the Microsoft Edge DevTools, open the **Local** tab. Your add-in will be listed by its name.

5. Select the add-in name to attach the debugger to your add-in. A new Microsoft Edge DevTools window will open when the debugger attaches to your add-in.

6. On the **Network** tab of the new window, select **Clear cache**.
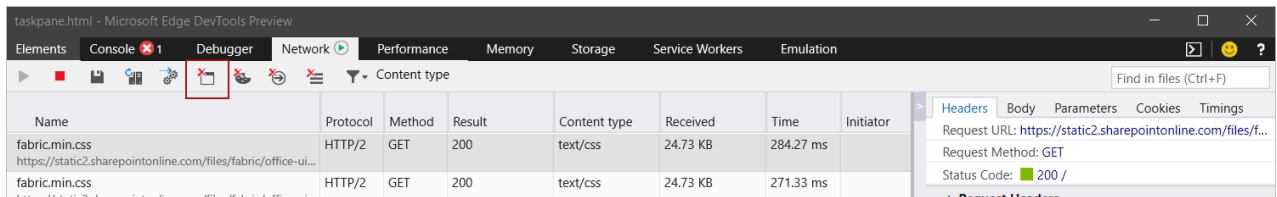


7. If completing these steps doesn't produce the desired result, try selecting **Always refresh from server**.
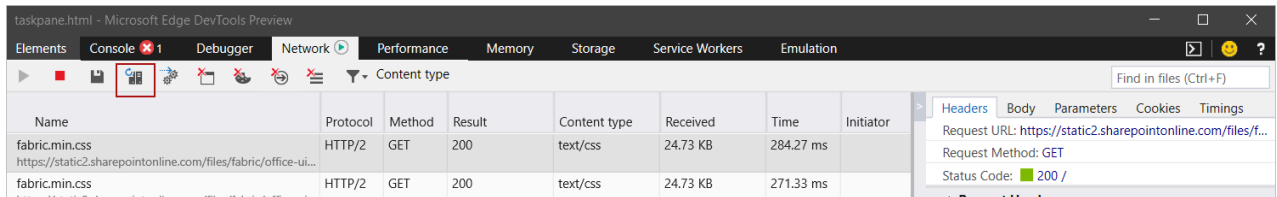


# Clear the Office cache on Mac

You can choose to clear the web or both the web and Wef caches on Mac.
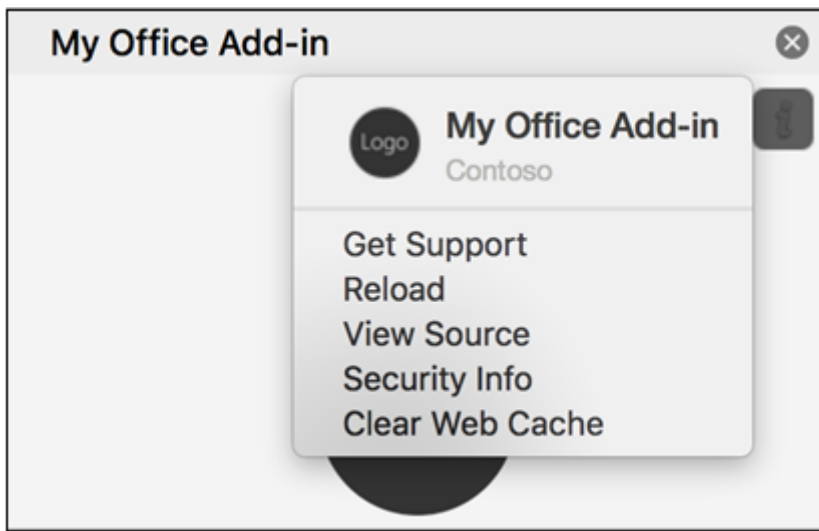
## Clear the web cache

Normally, the web cache is cleared by reloading the add-in. If more than one add-in exists in the same document, the process of automatically clearing the cache on reload might not be reliable.

To clear the web cache in Excel, PowerPoint, and Word, use the personality menu of any task pane add-in.

> ⓘ **Note**
>
> - The personality menu in task panes is only supported in Excel, PowerPoint, and Word. Because it isn't supported in Outlook, you must use the **option to clear both the web and Wef caches instead**.
> - The personality menu is only shown in macOS Version 10.13.6 or later.

From the add-in's task pane, choose the personality menu. Then, choose **Clear Web Cache**.

## Clear the web and Wef caches

To clear both the web and Wef caches on Mac, delete the contents of the `~/Library/Containers/com.Microsoft.OsfWebHost/Data/` and `~/Library/Containers/com.microsoft.{host}/Data/Documents/wef` folders. Replace `{host}` with the Office application, such as `Excel`.

> 💡 **Tip**
>
> Use the terminal or Finder to search for the specified folders. To look for these folders via Finder, you must set Finder to show hidden files. Finder displays the folders inside the **Containers** directory by product name, such as **Microsoft Excel** instead of **com.microsoft.Excel**.

Deleting the contents of the `~/Library/Containers/com.microsoft.{host}/Data/Documents/wef` folder removes all sideloaded add-ins from an application.

> ① **Note**
>
> If the `~/Library/Containers/com.Microsoft.OsfWebHost/Data/` folder doesn't exist, check for the following folders via terminal or Finder. If found, delete the contents of each folder.
>
> - `~/Library/Containers/com.microsoft.{host}/Data/Library/Caches/` where `{host}` is the Office application (e.g., `Excel`)
> - `~/Library/Containers/com.microsoft.{host}/Data/Library/Application Support/Microsoft/Office/16.0/Wef/` where `{host}` is the Office application (e.g., `Excel`)

- `~/Library/Containers/com.microsoft.Office365ServiceV2/Data/Caches/com.microsoft.Office365ServiceV2/`
- `~/Library/Containers/com.microsoft.Office365ServiceV2/Data/Library/Caches/com.microsoft.Office365ServiceV2/`

# Clear the Office cache on iOS

To clear the web cache on iOS, call `window.location.reload(true)` from JavaScript in the add-in. This forces the add-in to reload. Alternatively, reinstall Office.

# See also

- Troubleshoot development errors with Office Add-ins
- Debug add-ins using developer tools for Internet Explorer
- Debug add-ins using developer tools for Edge Legacy
- Debug add-ins using developer tools in Microsoft Edge (Chromium-based)
- Debug your add-in with runtime logging
- Sideload Office Add-ins for testing
- Office Add-ins manifest
- Validate an Office Add-in's manifest
- Uninstall add-ins under development