Manage both a unified manifest and an add-in only manifest version of your Office Add-in

Article • 06/10/2024

Microsoft is making a number of improvements to the Microsoft 365 developer platform. These improvements provide more consistency in the development, deployment, installation, and administration of all types of extensions of Microsoft 365, including Office Add-ins.

One important improvement we're working on is the ability to create a single unit of distribution for all your Microsoft 365 extensions by using the same manifest format: the JSON-formatted unified manifest for Microsoft 365.

> 💡 **Tip**
>
> For information about converting an existing add-in to use the unified manifest, see Convert an add-in to use the unified manifest for Microsoft 365.

After you've created a version of your add-in that uses the unified manifest, you must link the existing add-in and the app built using the unified manifest to ensure users don't see two copies of the add-in UI inside of Outlook.

> ⓘ **Note**
>
> The configuration described below doesn't take effect for up to 24 hours after the new version is installed on a user's computer. During that period, the UI of both versions is visible. For example, if both versions have a custom ribbon button, both buttons appear on the ribbon.

Use the following steps.

1. Open the extension object in the "extensions" array.
2. Create an "alternatives" array property, if there isn’t one already.
3. In the "alternatives" array, create an "alternate" object that has a "hide" property.
4. If the existing add-in is marketed through AppSource, give the "hide" object a "storeOfficeAddin" property. Otherwise, skip to step 6.

5. Give the "storeOfficeAddin" object two properties:

    - An "officeAddinId" with the GUID of the old add-in as its value.
    - An "assetId" with the AppSource asset ID as its value.

    The following is an example:

    JSON

    ```json
    "extensions": [
        ...
        {
            ...
            "alternates": [
                ...
                {
                    ...
                    "hide": {
                        "storeOfficeAddin": {
                            "officeAddinId": "b5a2794d-4aa5-4023-a84b-c60a3cbd33d4",
                            "assetId": "WA999999999"
                        }
                    }
                }
            ]
        }
    ]
    ```

    > **Note**
    >
    > - The asset ID of the add-in in your unified manifest must match with an existing add-in that has been published by your seller account on Partner Center. If the asset ID of the add-in that you have linked in your unified manifest doesn't match an existing offer published by your seller account, the unified manifest submission will fail. You'll need to update the manifest to use the correct add-in asset ID and re-submit the unified manifest.
    > - An existing add-in can only be hidden by a single unified manifest. At this time, you may not use multiple unified manifests to hide the same add-in. If you try to hide an already linked add-in using a different unified manifest, the submission will fail. You'll need to remove the linking and re-submit the unified manifest.

6. If the old add-in isn't distributed through AppSource, then give the "hide" object a "customOfficeAddin" property.

7. Give the "customOfficeAddin" object an "officeAddinId" property with the GUID of the old add-in as its value. The following is an example.

JSON

```json
"extensions": [
    ...
    {
        ...
        "alternates": [
            ...
            {
                ...
                "hide": {
                    "customOfficeAddin": {
                        "officeAddinId": "b5a2794d-4aa5-4023-a84b-
c60a3cbd33d4"
                    }
                }
            }
        ]
    }
]
```

Don't remove the existing add-in from AppSource or the Microsoft 365 Admin Center, or earlier versions of Office will no longer be able to use your add-in.

Maintain both versions for the immediate future

Generally, add-ins that use the unified manifest can be installed only on Microsoft 365 Version 2307 (Build 16626.20132) and later. However, there are two exceptions which enable these add-ins to be installed on older versions of Microsoft 365 and on perpetual license versions of Office.

- The user's Microsoft 365 administrator deploys the add-in for all users.
- The user installs the add-in on another Microsoft 365 client app that *is* version Version 2307 (Build 16626.20132) and later. This makes the add-in available on the same user's other Office clients, including older or perpetual license.

If you have users on older or perpetual license versions for which these exceptions don't apply, then you will need to maintain both versions of the add-in. When all of your users are working with Office versions that support the unified manifest, you can remove the XML version from deployment.

There are also some scenarios where you might want to maintain both both versions of the add-in for an extended period. For example, there are two features of add-ins that aren't supported with the unified manifest because they're little used or deprecated. You may choose to maintain a version of your add-in that uses these features. The following are the features that aren't supported in the unified manifest.

- Outlook modules aren't supported. But you can provide a nearly identical experience using the unified manifest by including a Teams Tab with your add-in in a single app.
- Outlook contextual add-ins (also known as "activation rules") aren't supported. But you can provide similar experiences using the unified manifest and Event-based activation.

The critical requirement for making two versions available is to be sure that the two of them appear distinct in the Outlook UI.

- Give the new version a different name from the existing add-in.
- Create and use different icons for the new version.
- Be sure that the "id" property of the unified manifest in the new version is a different GUID from the <Id> element in the add-in only manifest of the existing add-in.

> ⓘ **Note**
>
> If you use the same name and icon, the old and new solutions appear indistinguishable in the Outlook UI for add-in installation.