Deploy and publish Office Add-ins

08/15/2025

You can use one of several methods to deploy your Office Add-in for testing or distribution to users. The deployment method can also affect which platforms surface your add-in.

> **Note**
>
> For information about how end users acquire, insert, and run add-ins, see **Start using your Office Add-in** ↗.

Primary publication methods

The following table summarizes the primary publication methods that can be used regardless of which type of manifest the add-in uses. If the add-in uses the add-in only manifest, see also Additional publication methods for the add-in only manifest.

⛶ Expand table

Method	Use
Sideloading	As part of your development process, to test your add-in running on Windows, iPad, Mac, or in a browser. (Not for production add-ins.)
AppSource	To distribute your add-in publicly to users.
Integrated apps portal in the Microsoft 365 admin center	To distribute your add-in to users in your organization.

Production deployment methods

The following sections provide additional information about the deployment methods that are most commonly used to distribute production Office Add-ins to users.

AppSource

You can make your add-in available through AppSource ↗, Microsoft's online app store which is accessible through a browser and through the UI of Office applications. Distribution through AppSource gives you the option of including installation of your add-in with the installation of your Windows app or a COM or VSTO add-in. For more information, see Publish to your Office Add-in to AppSource.

> **⊙ Note**
>
> If you plan to __publish__ your add-in to AppSource and make it available within the Office experience, make sure that you conform to the __Commercial marketplace certification policies__. For example, to pass validation, your add-in must work across all platforms that support the methods that you define (for more information, see __section 1120.3__ and the __Office Add-in application and availability page__).

Integrated apps portal in the Microsoft 365 admin center

The Microsoft 365 admin center makes it easy for an administrator to deploy Office Add-ins to users and groups in their organization. Add-ins deployed via the admin center are available to users in their Office applications right away, with no client configuration required. You can use integrated apps portal to deploy internal add-ins as well as add-ins provided by independent software vendors (ISVs). The integrated apps portal also shows admins add-ins and other apps bundled together by same ISV, giving them exposure to the entire experience across the Microsoft 365 platform.

When you link your Office Add-ins, Teams apps, SharePoint Framework (SPFx) apps, and other apps together, you create a single software as a service (SaaS) offering for your customers. For general information about this process, see How to plan a SaaS offer for the commercial marketplace. For specifics on how to create the offer, see Create the offer.

For more information on the deployment process, see Get started with the integrated apps portal.

> **⊙ Note**
>
> If your add-in uses the __unified manifest for Microsoft 365__ and is distributed as an internal add-in in the integrated apps portal (instead of being acquired by the administrator from AppSource), it won't be installable by users with certain versions of Office. For more information, see __Office Add-ins with the unified app manifest for Microsoft 365 - Client and platform support__.

> **ⓘ Important**
>
> Customers in sovereign or government clouds don't have access to the integrated apps portal. They use Centralized Deployment instead. (See __Additional publication methods for the add-in only manifest__ later in this article.) Centralized Deployment is a similar deployment method, but doesn't expose connected add-ins and apps to the admin. For

more information, see **Determine if Centralized Deployment of add-ins works for your organization**.

Deploy updates

When you add features or fix bugs in your add-in, you'll need to deploy the updates. If your add-in is deployed by one or more admins to their organizations, some manifest changes will require the admin to consent to the updates. Users remain on the existing version of the add-in until the admin consents to the updates. The following manifest changes will require the admin to consent again.

- Changes to requested permissions. See Requesting permissions for API use in add-ins and Understanding Outlook add-in permissions.
- Additional or changed Scopes. (Not applicable if the add-in uses the unified manifest for Microsoft 365.)
- Additional or changed Events.

> ⓘ **Note**
>
> Whenever you make a change to the manifest, you must raise the version number of the manifest.
>
> - If the add-in uses the add-in only manifest, see **Version element**.
> - If the add-in uses the unified manifest, see **version property**.

Additional publication methods for the add-in only manifest

The following table summarizes publication methods that are available *only when the add-in uses the add-in only manifest.*

⛶ Expand table

Method	Use	Support limitations
Network share	As part of your development process, to test your add-in running on Windows computers other than your development computer after you have published the add-in to a server other than localhost.	- Not supported for production add-ins. - Not supported for Outlook add-ins. - Not supported for testing on iPad, Mac, or the web.

Method	Use	Support limitations
SharePoint catalog	In an on-premises environment, to distribute your add-in to users in your organization.	• Not supported for Outlook add-ins. • Not supported for Office on Mac. • Not supported for add-ins with any feature that requires a `<VersionOverrides>` element in the add-in only manifest.
Exchange server	In an on-premises or online environment, to distribute Outlook add-ins to users.	Only supported for Outlook add-ins.
Centralized Deployment	To distribute your add-in to users in your organization.	

SharePoint app catalog deployment

A SharePoint app catalog is a special SharePoint site collection that you can create to host the manifests (add-in only manifest type) of a Word, Excel, or PowerPoint add-in. If you're deploying add-ins in an on-premises environment and none of the add-in users use a Mac, consider using a SharePoint catalog. For details, see Publish task pane and content add-ins to a SharePoint catalog.

Because SharePoint catalogs don't support new add-in features implemented in the `VersionOverrides` node of the manifest, including add-in commands, for these add-ins, we recommend that you use Centralized Deployment via the admin center if possible.

Outlook add-in Exchange server deployment

For on-premises and online environments that don't use the Microsoft Entra identity service, you can deploy Outlook add-ins via the Exchange server.

Outlook add-in deployment requires:

- Microsoft 365, Exchange Online, or Exchange Server 2016 or later
- Outlook 2016 or later

To assign and manage add-ins for your tenants and users, use Exchange PowerShell. For more information, see Add-ins for Outlook in Exchange Server and Add-ins for Outlook in Exchange Online.

It's important to note that some versions of Outlook clients and Exchange servers may only support certain Mailbox requirement sets. For details about supported requirement sets, see Requirement sets supported by Exchange servers and Outlook clients.

GoDaddy Microsoft 365 SKUs

[Microsoft 365 subscriptions provided by GoDaddy](#) ↗ have limited support for add-ins. The following options are **not** supported.

- Deployment through the Microsoft 365 admin center.
- Deployment through Exchange servers.
- Acquiring add-ins from AppSource.

See also

- [Sideload Outlook add-ins for testing](#)
- [Publish to your Office Add-in to AppSource](#)
- [AppSource](#) ↗
- [Design guidelines for Office Add-ins](#)
- [Create effective AppSource listings](#)
- [Troubleshoot user errors with Office Add-ins](#)
- [What is the Microsoft commercial marketplace?](#)
- [Microsoft Dev Center app publishing page](#)