Special requirements for add-ins on the iPad

07/30/2025

You'll need to make additional considerations if you want to make your add-in available on iPad. If your add-in uses only Office APIs that are supported on iPad, customers can install it on their devices. However, if you're publishing to AppSource, there are some additional requirements you'll need to meet.

For details on API compatibility, see Specify Office applications and API requirements.

iPad AppSource requirements

Here's what you need to know when submitting your add-in to AppSource for iPad users.

Expand table

Task	Description	Resources
Apply iOS design best practices.	Make your add-in UI feel native to iOS by following Apple's design guidelines.	Designing for iOS
Make your add-in free.	Your add-in must be free on iPad. Office on iPad is a great way to reach new users who might become customers on other platforms.	Certification policy 1120.2
Remove commerce features on iPad.	When running on iPad, your add-in can't include in-app purchases, trial offers, upselling UI, or links to online stores. Your Privacy Policy and Terms of Use pages must also be commerce-free. You can still have commerce on other platforms. Check the Office.context.commerceAllowed property and hide commerce features when it returns `false`.	Certification policy 1100.3
Submit to AppSource correctly.	In Partner Center, go to the **Product setup** page and select **Make my product available on iOS and Android (if applicable)**. You'll also need to provide your Apple developer ID in Account settings. Don't forget to review the Application Provider Agreement.	Make your solutions available in AppSource and within Office

Detecting iPad devices

Want to provide a different experience for iPad users? Your add-in can detect what device it's running on and adjust accordingly. Use the Office.context.touchEnabled and

Office.context.commerceAllowed properties to detect iPad devices.

JavaScript

```
const isTouchEnabled = Office.context.touchEnabled;
const allowCommerce = Office.context.commerceAllowed;

// On iPad, touchEnabled returns true and commerceAllowed returns false.
if (isTouchEnabled && !allowCommerce) {
    // Likely running on iPad - implement the iPad-specific UI.
    enableIPadInterface();
    hideCommerceFeatures();
}
```

iPad development best practices

Ready to start building for iPad? Here are the key practices that'll help you succeed.

Develop and debug on Windows or Mac, then test on iPad

You can't develop directly on an iPad. Develop and debug your add-in on a Windows or Mac computer, then sideload it to an iPad for testing. Since Office add-ins use the same APIs across platforms (Windows, Mac, and iOS), your code should work consistently everywhere.

For step-by-step guidance, see:

- Test and debug Office Add-ins
- Sideload Office Add-ins on iPad for testing

Handle API compatibility with manifest requirements or runtime checks

You have two ways to ensure your add-in works properly across different Office versions.

Option 1: Specify requirements in your manifest. When you declare API requirements in your add-in's manifest, Office automatically checks if the client app supports those APIs. If they're available, your add-in will be available too.

Option 2: Use runtime checks. Alternatively, you can check if specific methods are available before using them in your code. This approach ensures your add-in is always available and provides additional functionality when supported APIs are present.

For detailed guidance on both approaches, see Specify Office applications and API requirements.

Outlook add-ins on iPad

For information about designing Outlook add-ins that look good and work well in Outlook on mobile devices, see Add-ins for Outlook on mobile devices.

> **Note**
>
> If you're using Fluent UI React for your design elements, many of these elements are built into the design system.