Office Add-in design language

Article • 02/11/2025

The Office design language is a clean and simple visual system that ensures consistency across experiences. It contains a set of visual elements that define Office interfaces, including:

- A standard typeface
- A common color palette
- A set of typographic sizes and weights
- Icon guidelines
- Shared icon assets
- Animation definitions
- Common components

Fluent UI is the official front-end framework for building with the Office design language. Using Fluent UI is optional, but it's the fastest way to ensure that your add-ins feel like a natural extension of Office. Take advantage of Fluent UI to design and build add-ins that complement Office.

Many Office Add-ins are associated with a preexisting brand. You can retain a strong brand and its visual or component language in your add-in. Look for opportunities to retain your own visual language while integrating with Office. Consider ways to swap out Office colors, typography, icons, or other stylistic elements with elements of your own brand. Consider ways to follow common add-in layouts or UX design patterns while inserting controls and components that are familiar to your customers.

Inserting a heavily branded HTML-based UI inside of Office can create dissonance for customers. Find a balance that fits seamlessly in Office but also clearly aligns with your service or parent brand. When an add-in doesn't fit with Office, it's often because stylistic elements conflict. For example, typography is too large and off grid, colors are contrasting or particularly loud, or animations are superfluous and behave differently than Office. The appearance and behavior of controls or components veer too far from Office standards.

Review guidelines for visual elements

Learn about each visual element that makes up an Office Add-in, including guidelines and recommended practices.

- Color guidelines for Office Add-ins

- Icon guidelines for Office Add-ins
- Layout guidelines for Office Add-ins
- Using motion in Office Add-ins
- Typography guidelines for Office Add-ins

Design toolkits for download

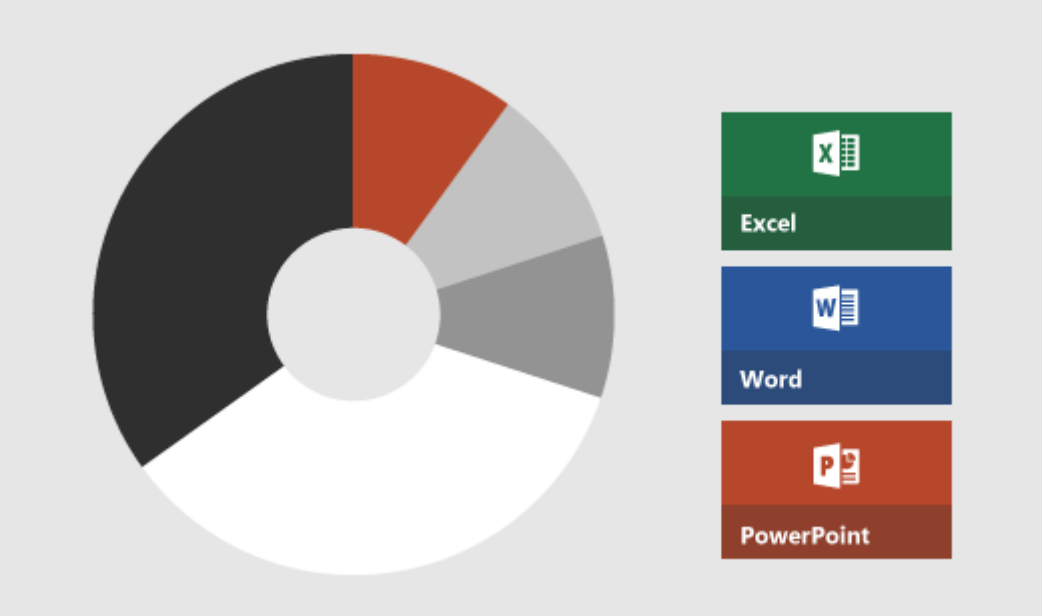
To help you get started, we've created toolkits for use with either the Sketch application for Mac or the Adobe XD application for Windows or Mac. The following downloads include all of our available patterns, along with brief descriptions and layout recommendations.

- Fluent UI Design Sketch Toolkit
- Fluent UI Design Adobe XD Toolkit
- Add-in Sketch Toolkit
- Add-in Adobe XD Toolkit
- Segoe UI and Fabric MDL2 icon font

Color guidelines for Office Add-ins

Article • 06/20/2024

Color is often used to emphasize brand and reinforce visual hierarchy. It helps identify an interface as well as guide customers through an experience. Inside Office, color is used for the same goals but it's applied purposefully and minimally. At no point does it overwhelm customer content. Even when each Office app is branded with its own dominant color, it's used sparingly.



Fluent UI React and Fabric Core include a set of default theme colors. When these libraries are applied to the components or layouts of an Office Add-in, the same goals apply. Color should communicate hierarchy, purposefully guiding customers to action without interfering with content. Theme colors from Fluent UI React or Fabric Core can introduce a new accent color to the overall interface. These accent colors can conflict with Office app branding and the hierarchy. Consider ways to avoid conflicts and interference. Use neutral accents or overwrite theme colors to match Office app branding or your own brand colors.

In Office applications, customers personalize their interfaces by applying an Office UI theme. Customers choose between four UI themes to vary styling of backgrounds and buttons in Excel, Outlook, PowerPoint, Word, and other apps in the Office suite. To make your add-ins feel like a natural part of Office and respond to personalization, use our Theming APIs. For example, task pane background colors switch to a dark gray in some themes. With the Theming APIs, follow suit and adjust foreground text to ensure accessibility.

Tip

- To ensure that your add-in applies the correct color combinations in its UI, test it with all four Office themes, including the **Use system setting** option.
- For guidance on how to dynamically match the design of your PowerPoint add-in with the current document or Office theme, see **Use document themes in your PowerPoint add-ins**.

Apply the following general guidelines for color.

- Use color sparingly to communicate hierarchy and reinforce brand.
- Overuse of a single accent color applied to both interactive and non-interactive elements can lead to confusion. For example, avoid using the same color for selected and unselected items in a navigation menu.
- Avoid unnecessary conflicts with Office branded app colors.
- Use your own brand colors to build association with your service or company.
- Ensure that all text is accessible. Be sure that there is a 4.5:1 contrast ratio between foreground text and background.
- Be aware of color blindness. Use more than just color to indicate interactivity and hierarchy.
- To learn more about designing add-in command icons with the Office icon color palette, see icon guidelines.

Icons

Article • 04/30/2024

Icons are the visual representation of a behavior or concept. They are often used to add meaning to controls and commands. Visuals, either realistic or symbolic, enable the user to navigate the UI the same way signs help users navigate their environment. They should be simple, clear, and contain only the necessary details to enable customers to quickly parse what action will occur when they choose a control.

Office app ribbon interfaces have a standard visual style. This ensures consistency and familiarity across Office apps. The guidelines will help you design a set of PNG assets for your solution that fit in as a natural part of Office.

Many HTML containers contain controls with iconography. Use Fabric Core's custom font to render Office styled icons in your add-in. The icon font provided by Fabric Core contains many glyphs for common Office metaphors that you can scale, color, and style to suit your needs. If you have an existing visual language with your own set of icons, feel free to use it in your HTML canvases. Building continuity with your own brand with a standard set of icons is an important part of any design language. Be careful to avoid creating confusion for customers by conflicting with Office metaphors.

Design icons for add-in commands

Add-in commands add buttons, text, and icons to the Office UI. Your add-in command buttons should provide meaningful icons and labels that clearly identify the action the user is taking when they use a command. The following articles provide stylistic and production guidelines to help you design icons that integrate seamlessly with Office.

- For the Monoline style of Microsoft 365, see Monoline style icon guidelines for Office Add-ins.
- For the Fresh style of perpetual Office 2016 and later, see Fresh style icon guidelines for Office Add-ins.

> **Note**
>
> You must choose one style or the other and your add-in will use the same icons whether it's running in Microsoft 365 or perpetual Office.

See also

- Add-in development best practices
- Add-in commands for Excel, Word, and PowerPoint

Collaborate with us on GitHub

The source for this content can be found on GitHub, where you can also create and review issues and pull requests. For more information, see our contributor guide.

Office Add-ins feedback

Office Add-ins is an open source project. Select a link to provide feedback:

Open a documentation issue

Provide product feedback

Fresh style icon guidelines for Office Add-ins

08/25/2025

Perpetual Office 2016 and later use Microsoft's Fresh style iconography. If you would prefer that your icons match the Monoline style of Microsoft 365, see Monoline style icon guidelines for Office Add-ins.

Office Fresh visual style

The Fresh icons include only essential communicative elements. Non-essential elements including perspective, gradients, and light source are removed. The simplified icons support faster parsing of commands and controls. Follow this style to best fit with Office perpetual clients.

Best practices

Follow these guidelines when you create your icons.

Expand table

Do	Don't
Keep visuals simple and clear, focusing on the key elements of the communication.	Don't use artifacts that make your icon look messy.
Use the Office icon language to represent behaviors or concepts.	Don't repurpose Fabric Core glyphs for add-in commands in the Office app ribbon or contextual menus. Fabric Core icons are stylistically different and won't match.
Reuse common Office visual metaphors such as paintbrush for format or magnifying glass for find.	Don't reuse visual metaphors for different commands. Using the same icon for different behaviors and concepts can cause confusion.
Redraw your icons to make them small or larger. Take the time to redraw cutouts, corners, and rounded edges to maximize line clarity.	Don't resize your icons by shrinking or enlarging in size. This can lead to poor visual quality and unclear actions. Complex icons created at a larger size may lose clarity if resized to be smaller without redraw.

Do	Don't
Use a white fill for accessibility. Most objects in your icons will require a white background to be legible across Office UI themes and in high-contrast modes.	Avoid relying on your logo or brand to communicate what an add-in command does. Brand marks aren't always recognizable at smaller icon sizes and when modifiers are applied. Brand marks often conflict with Office app ribbon icon styles, and can compete for user attention in a saturated environment.
Use the PNG format with a transparent background.	*None*
Avoid localizable content in your icons, including typographic characters, indications of paragraph rags, and question marks.	*None*

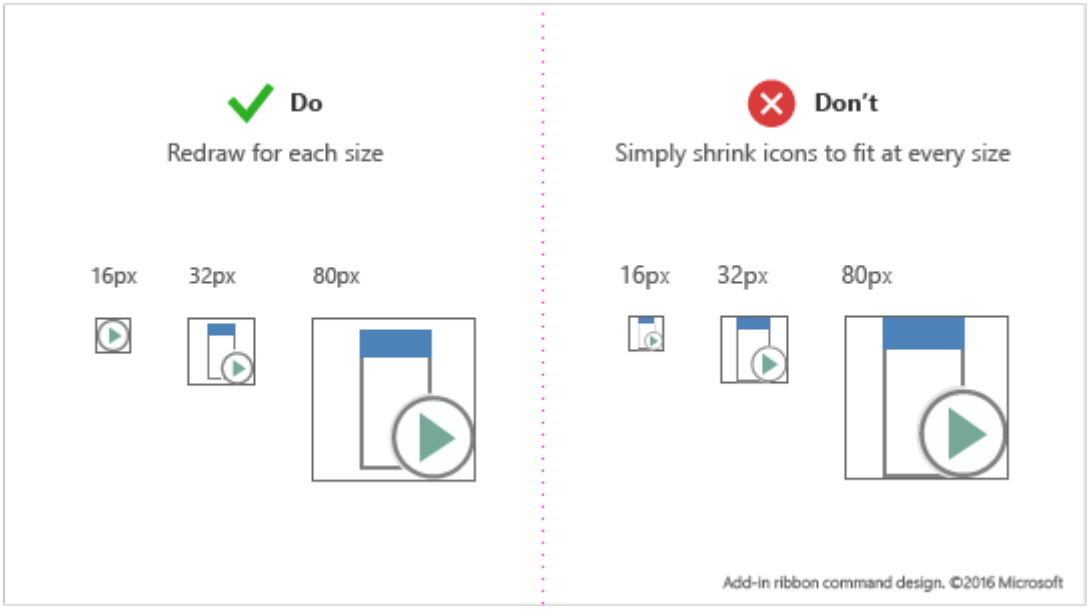
Icon size recommendations and requirements

Office desktop icons are bitmap images. Different sizes will render depending on the user's DPI setting and touch mode. Include all eight supported sizes to create the best experience in all supported resolutions and contexts. The following are the supported sizes - three are required.

- 16 px (Required)
- 20 px
- 24 px
- 32 px (Required)
- 40 px
- 48 px
- 64 px (Recommended, best for Mac)
- 80 px (Required)

> ⓘ **Important**
>
> For an image that is your add-in's representative icon, see Create effective listings in AppSource and within Office for size and other requirements.

Make sure to redraw your icons for each size rather than shrink them to fit.

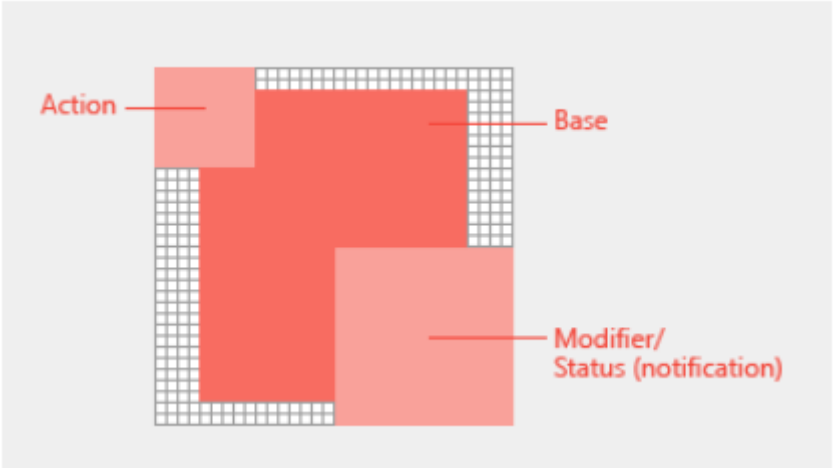


Icon anatomy and layout

Office icons are typically comprised of a base element with action and conceptual modifiers overlaid. Action modifiers represent concepts such as add, open, new, or close. Conceptual modifiers represent status, alteration, or a description of the icon.

To create commands that align with the Office UI, follow layout guidelines for the base element and modifiers. This ensures that your commands look professional and that your customers will trust your add-in. If you make exceptions to these guidelines, do so intentionally.

The following image shows the layout of base elements and modifiers in an Office icon.



- Center base elements in the pixel frame with empty padding all around.
- Place action modifiers on the top left.
- Place conceptual modifiers on the bottom right.
- Limit the number of elements in your icons. At 32 px, limit the number of modifiers to a maximum of two. At 16 px, limit the number of modifiers to one.

Base element padding

Place base elements consistently across sizes. If base elements can't be centered in the frame, align them to the top left, leaving the extra pixels on the bottom right. For best results, apply the padding guidelines listed in the table in the following section.

Modifiers

All modifiers should have a 1 px transparent cutout between each element, including the background. Elements shouldn't directly overlap. Create whitespace between rules and edges. Modifiers can vary slightly in size, but use these dimensions as a starting point.

Expand table

Icon size	Padding around base element	Modifier size
16 px	0	9 px
20 px	1px	10 px
24 px	1px	12 px
32 px	2px	14 px
40 px	2px	20 px
48 px	3px	22 px
64 px	5px	29 px
80 px	5px	38 px

Icon colors

> **Note**
>
> These color guidelines are for ribbon icons used in Add-in commands. These icons aren't rendered with Fluent UI.

Office icons have a limited color palette. Use the colors listed in the following table to guarantee seamless integration with the Office UI. Apply the following guidelines to the use of color.

- Use color to communicate meaning rather than for embellishment. It should highlight or emphasize an action, status, or an element that explicitly differentiates the mark.
- If possible, use only one additional color beyond gray. Limit additional colors to two at the most.
- Colors should have a consistent appearance in all icon sizes. Office icons have slightly different color palettes for different icon sizes. 16 px and smaller icons are slightly darker and more vibrant than 32 px and larger icons. Without these subtle adjustments, colors appear to vary across sizes.

Expand table

Color name	RGB	Hex	Color	Category
Text Gray (80)	80, 80, 80	#505050		Text
Text Gray (95)	95, 95, 95	#5F5F5F		Text
Text Gray (105)	105, 105, 105	#696969		Text
Dark Gray 32	128, 128, 128	#808080		32 px and above
Medium Gray 32	158, 158, 158	#9E9E9E		32 px and above
Light Gray ALL	179, 179, 179	#B3B3B3		All sizes
Dark Gray 16	114, 114, 114	#727272		16 px and below
Medium Gray 16	144, 144, 144	#909090		16 and below
Blue 32	77, 130, 184	#4d82B8		32 px and above

Color name	RGB	Hex	Color	Category
Blue 16	74, 125, 177	#4A7DB1		16 px and below
Yellow ALL	234, 194, 130	#EAC282		All sizes
Orange 32	231, 142, 70	#E78E46		32 px and above
Orange 16	227, 142, 70	#E3751C		16 px and below
Pink ALL	230, 132, 151	#E68497		All sizes
Green 32	118, 167, 151	#76A797		32 px and above
Green 16	104, 164, 144	#68A490		16 px and below
Red 32	216, 99, 68	#D86344		32 px and above
Red 16	214, 85, 50	#D65532		16 px and below
Purple 32	152, 104, 185	#9868B9		32 px and above
Purple 16	137, 89, 171	#8959AB		16 px and below

Icons in high contrast modes

Office icons are designed to render well in high contrast modes. Foreground elements are well differentiated from backgrounds to maximize legibility and enable recoloring. In high contrast modes, Office will recolor any pixel of your icon with a red, green, or blue value less than 190 to full black. All other pixels will be white. In other words, each RGB channel is assessed where 0-189 values are black and 190-255 values are white. Other high-contrast themes recolor using the same 190 value threshold but with different rules. For example, the high-contrast white theme will recolor all pixels greater than 190 opaque but all other pixels as transparent. Apply the following guidelines to maximize legibility in high-contrast settings.

- Aim to differentiate foreground and background elements along the 190 value threshold.
- Follow Office icon visual styles.
- Use colors from our icon palette.
- Avoid the use of gradients.
- Avoid large blocks of color with similar values.

See also

Unified manifest reference

- "extensions.ribbons" array

Add-in only manifest reference

- Icon manifest element
- IconUrl manifest element
- HighResolutionIconUrl manifest element
- Create an icon for your add-in

Monoline style icon guidelines for Office Add-ins

Article • 02/12/2025

Monoline style iconography are used in Office apps. If you'd prefer that your icons match the Fresh style of perpetual Office 2016 and later, see Fresh style icon guidelines for Office Add-ins.

Office Monoline visual style

The goal of the Monoline style to have consistent, clear, and accessible iconography to communicate action and features with simple visuals, ensure the icons are accessible to all users, and have a style that is consistent with those used elsewhere in Windows.

The following guidelines are for 3rd party developers who want to create icons for features that will be consistent with the icons already present Office products.

Design principles

- Simple, clean, clear.
- Contain only necessary elements.
- Inspired by Windows icon style.
- Accessible to all users.

Convey meaning

- Use descriptive elements such as a page to represent a document or an envelope to represent mail.
- Use the same element to represent the same concept. For example, mail is always represented by an envelope, not a stamp.
- Use a core metaphor during concept development.

Reduction of elements

- Reduce the icon to its core meaning, using only elements that are essential to the metaphor.
- Limit the number of elements in an icon to two, regardless of icon size.

Consistency

Sizes, arrangement, and color of icons should be consistent.

Styling

Perspective

Monoline icons are forward-facing by default. Certain elements that require perspective and/or rotation, such as a cube, are allowed, but exceptions should be kept to a minimum.

Embellishment

Monoline is a clean minimal style. Everything uses flat color, which means there are no gradients, textures, or light sources.

Designing

Sizes

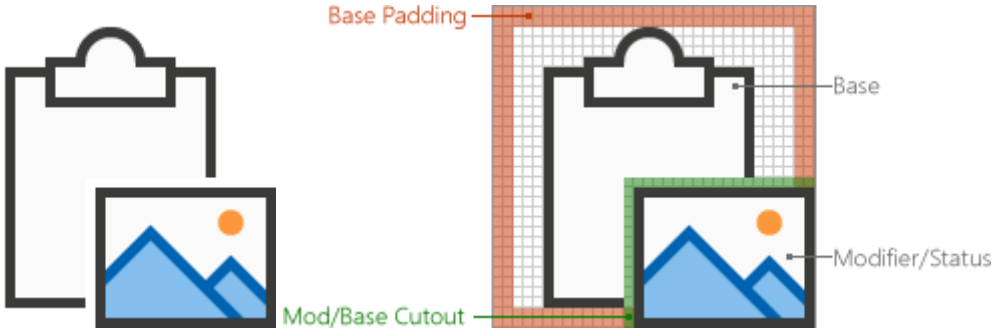
We recommend that you produce each icon in all these sizes to support high DPI devices. The absolutely *required* sizes are 16 px, 20 px, and 32 px, as those are the 100% sizes.

16 px, 20 px, 24 px, 32 px, 40 px, 48 px, 64 px, 80 px, 96 px

> ⓘ **Important**
>
> For an image that is your add-in's representative icon, see Create effective listings in AppSource and within Office for size and other requirements.

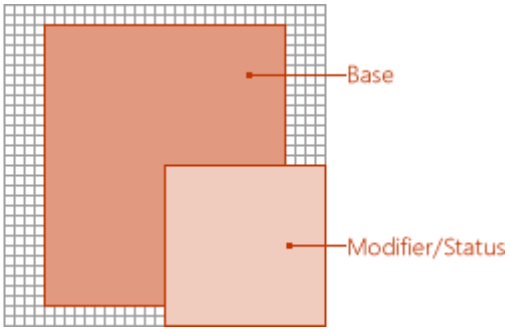
Layout

The following is an example of icon layout with a modifier.



Elements

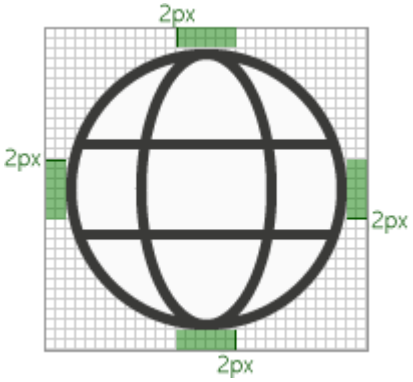
- **Base**: The main concept that the icon represents. This is usually the only visual needed for the icon, but sometimes the main concept can be enhanced with a secondary element, a modifier.
- **Modifier** Any element that overlays the base; that is, a modifier that typically represents an action or a status. It modifies the base element by acting as an addition, alteration, or a descriptor.



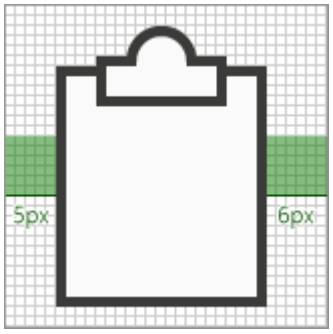
Construction

Element placement

Base elements are placed in the center of the icon within the padding. If it can't be placed perfectly centered, then the base should err to the top right. In the following example, the icon is perfectly centered.



In the following example, the icon is erring to the left.



Modifiers are almost always placed in the bottom right corner of the icon canvas. In some rare cases, modifiers are placed in a different corner. For example, if the base element would be unrecognizable with the modifier in the bottom right corner, then consider placing it in the upper left corner.

Padding

Each size icon has a specified amount of padding around the icon. The base element stays within the padding, but the modifier should butt up to the edge of the canvas, extending outside of the padding to the edge of the icon border. The following images show the recommended padding to use for each of the icon sizes.

Expand table

16px	20px	24px	32px	40px	48px	64px	80px	96px
0px	1px	1px	2px	2px	3px	4px	5px	6px

Line weights

Monoline is a style dominated by line and outlined shapes. Depending on what size you are producing the icon should use the following line weights.

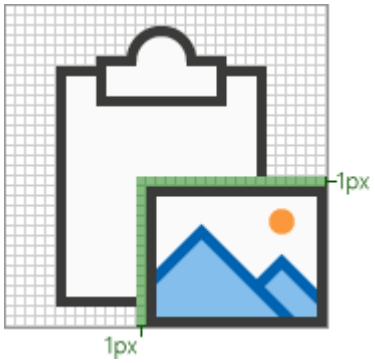
Expand table

Icon Size:	16px	20px	24px	32px	40px	48px	64px	80px	96px
Line Weight:	1px	1px	1px	1px	2px	2px	2px	2px	3px
Example icon:									

Cutouts

When an icon element is placed on top of another element, a cutout (of the bottom element) is used to provide space between the two elements, mainly for readability purposes. This usually happens when a modifier is placed on top of a base element, but there are also cases where neither of the elements is a modifier. These cutouts between the two elements is sometimes referred to as a "gap".

The size of the gap should be the same width as the line weight used on that size. If making a 16 px icon, the gap width would be 1px and if it's a 48 px icon then the gap should be 2px. The following example shows a 32 px icon with a gap of 1px between the modifier and the underlying base.



In some cases, the gap can be increase by a 1/2 px if the modifier has a diagonal or curved edge and the standard gap doesn't provide enough separation. This will likely only affect the icons with 1px line weight: 16 px, 20 px, 24 px, and 32 px.

Background fills

Most icons in the Monoline icon set require background fills. However, there are cases where the object would not naturally have a fill, so no fill should be applied. The following icons have a white fill.

The following icons have no fill. (The gear icon is included to show that the center hole isn't filled.)

Best practices for fills

Do

- Fill any element that has a defined boundary, and would naturally have a fill.
- Use a separate shape to create the background fill.
- Use **Background Fill** from the color palette.
- Maintain the pixel separation between overlapping elements.
- Fill between multiple objects.

Don't

- Don't fill objects that would not naturally be filled; for example, a paperclip.
- Don't fill brackets.
- Don't fill behind numbers or alpha characters.

Color

The color palette has been designed for simplicity and accessibility. It contains 4 neutral colors and two variations for blue, green, yellow, red, and purple. Orange is intentionally not included in the Monoline icon color palette. Each color is intended to be used in specific ways as outlined in this section.

Palette

Dark Gray – Standalone/Outline 58,58,56 #3A3A38		Background Fill 250,250,250 #FAFAFA
Medium Gray – Outline/Content 121,119,116 #797774		Light Gray - Fill 200,198,196 #C8C6C4

Blue – Standalone 30,139,205 # 1E8BCD	Blue – Outline 0,99,177 #0063B1	Blue - Fill 131, 190, 236 #83BEEC
Green - Standalone 24,171,80 #18AB50	Green - Outline 48,144,72 #309048	Green - Fill 161, 221, 170 #A1DDAA
Yellow - Standalone 251,152,59 #FB983B	Yellow - Outline 237, 135, 51 #ED8733	Yellow - Fill 248, 219, 143 #F8DB8F
Red - Standalone 237,61,59 #ED3D3B	Red - Outline 212, 35, 20 #D42314	Red - Fill 255, 145, 152 #FF9198
Purple - Standalone 168,70,178 #A846B2	Purple - Outline 146, 46, 155 #922E9B	Purple - Fill 212, 146, 216 #D492D8

How to use color

In the Monoline color palette, all colors have Standalone, Outline, and Fill variations. Generally, elements are constructed with a fill and a border. The colors are applied in one of the following patterns.

- The Standalone color alone for objects that have no fill.
- The border uses the Outline color and the fill uses the Fill color.
- The border uses the Standalone color and the fill uses the Background Fill color.

The following are examples of using color.

The most common situation will be to have an element use Dark Gray Standalone with Background Fill.

When using a colored Fill, it should always be with its corresponding Outline color. For example, Blue Fill should only be used with Blue Outline. But there are two exceptions to this general rule.

- Background Fill can be used with any color Standalone.
- Light Gray Fill can be used with two different Outline colors: Dark Gray or Medium Gray.

When to use color

Color should be used to convey the meaning of the icon rather than for embellishment. It should **highlight the action** to the user. When a modifier is added to a base element that has color, the base element is typically turned into Dark Gray and Background Fill so that the modifier can be the element of color, such as the case below with the "X" modifier being added to the picture base in the leftmost icon of the following set.

You should limit your icons to **one** additional color, other than the Outline and Fill mentioned above. However, more colors can be used if it's vital for its metaphor, with a limit of two additional colors other than gray. In rare cases, there are exceptions when more colors are needed. The following are good examples of icons that use just one color.

But the following icons use too many colors.

Use **Medium Gray** for interior "content", such as grid lines in an icon of a spreadsheet. Additional interior colors are used when the content needs to show the behavior of the control.

Text lines

When text lines are in a "container" (for example, text on a document), use medium gray. Text lines not in a container should be **Dark Gray**.

Text

Avoid using text characters in icons. Since Office products are used around the world, we want to keep icons as language neutral as possible.

Production

Icon file format

The final icons should be saved as .png image files. Use PNG format with a transparent background and have 32-bit depth.

See also

Unified manifest reference

- "extensions.ribbons" array

Add-in only manifest reference

- Icon manifest element
- IconUrl manifest element
- HighResolutionIconUrl manifest element
- Create an icon for your add-in

Layout

Article • 08/18/2023

Each HTML container embedded in Office will have a layout. These layouts are the main screens of your add-in. In them you'll create experiences that enable customers to initiate actions, modify settings, view, scroll, or navigate content. Design your add-in with a consistent layouts across screens to guarantee continuity of experience. If you have an existing website that your customers are familiar with using, consider reusing layouts from your existing web pages. Adapt them to fit harmoniously within Office HTML containers.

For guidelines on layout, see Task pane, Content. For more information about how to assemble Fluent UI React, or Office UI Fabric JS, components into common layouts and user experience flows, see UX design patterns templates.

Apply the following general guidelines for layouts.

- Avoid narrow or wide margins on your HTML containers. 20 pixels is a great default.
- Align elements intentionally. Extra indents and new points of alignment should aid visual hierarchy.
- Office interfaces are on a 4px grid. Aim to keep your padding between elements at multiples of 4.
- Overcrowding your interface can lead to confusion and inhibit ease of use with touch interactions.
- Keep layouts consistent across screens. Unexpected layout changes look like visual bugs that contribute to a lack of confidence and trust with your solution.
- Follow common layout patterns. Conventions help users understand how to use an interface.
- Avoid redundant elements like branding or commands.
- Consolidate controls and views to avoid requiring too much mouse movement.
- Create responsive experiences that adapt to HTML container widths and heights.

Using motion in Office Add-ins

Article • 06/29/2023

When you design an Office Add-in, you can use motion to enhance the user experience. UI elements, controls, and components often have interactive behaviors that require transitions, motion, or animation. Common characteristics of motion across UI elements define the animation aspects of a design language.

Because Office is focused on productivity, the animation language supports the goal of helping customers get things done. It strikes a balance between performant response, reliable choreography, and detailed delight. Office Add-ins sit within this existing animation language. Given this context, it's important to consider the following guidelines when applying motion.

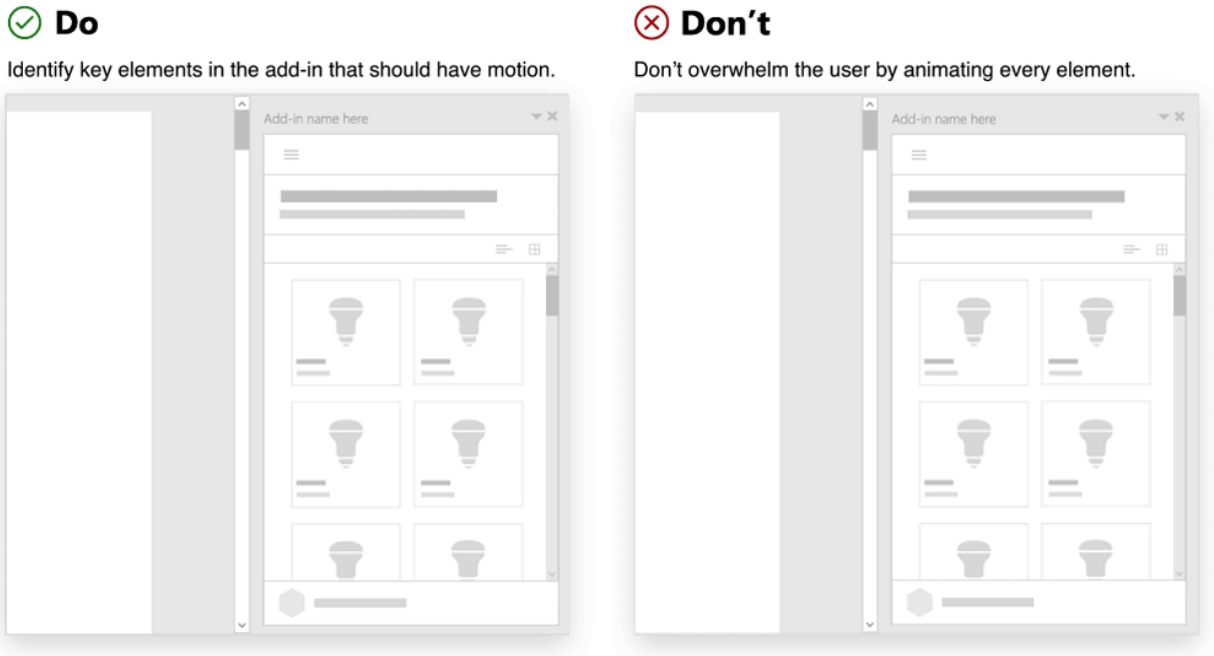
Create motion with a purpose

Motion should have a purpose that communicates additional value to the user. Consider the tone and purpose of your content when choosing animations. Handle critical messages differently than exploratory navigation.

Standard elements used in an add-in can incorporate motion to help focus the user, show how elements relate to each other, and validate user actions. Choreograph elements to reinforce hierarchy and mental models.

Best practices

Do	Don't
Identify key elements in the add-in that should have motion. Commonly animated elements in an add-in are panels, overlays, modals, tool tips, menus, and teaching call outs.	Don't overwhelm the user by animating every element. Avoid applying multiple motions that attempt to lead or focus the user on many elements at once.
Use simple, subtle motion that behaves in expected ways. Consider the origin of your triggering element. Use motion to create a link between the action and the resulting UI.	Don't create wait time for a motion. Motion in add-ins should not hinder task completion.



Use expected motions

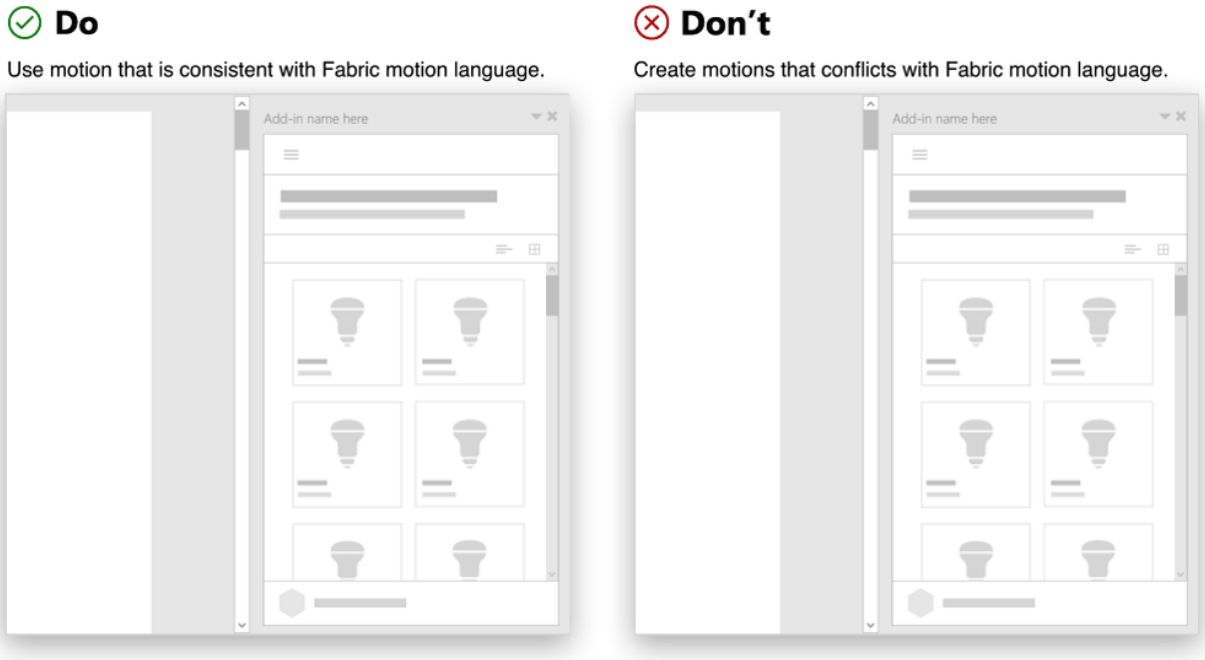
We recommend using Fluent UI to create a visual connection with the Office platform.

- Fluent UI React motion patterns
- Fabric Core motion and animation patterns

Use it to fit seamlessly in your add-in. It will help you create experiences that are more felt than observed. The animation CSS classes provide directionality, enter/exit, and duration specifics that reinforce Office mental models and provide opportunities for customers to learn how to interact with your add-in.

Best practices

Do	Don't
Use motion that aligns with behaviors in Fluent UI.	Don't create motions that interfere or conflict with common motion patterns in Office.
Ensure that there is a consistent application of motion across like elements.	Don't use different motions to animate the same component or object.
Create consistency with use of direction in animation. For example, a panel that opens from the right should close to the right.	Don't animate an element using multiple directions.

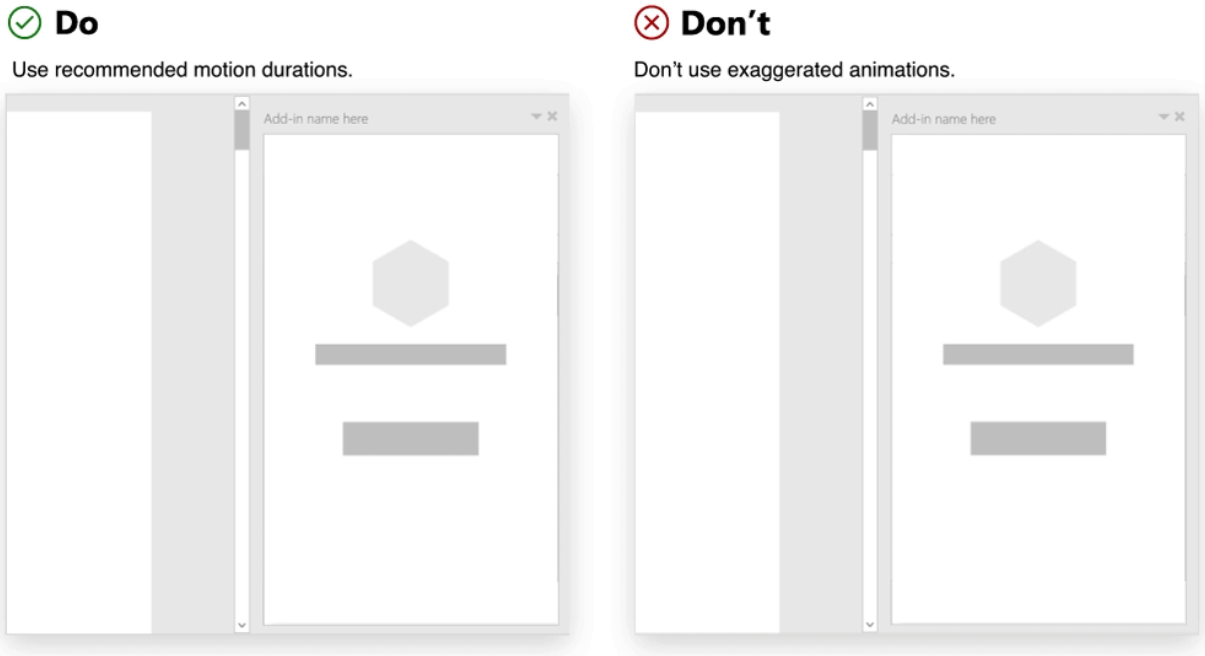


Avoid out of character motion for an element

Consider the size of the HTML canvas (task pane, dialog box, or content add-in) when implementing motion. Avoid overloading in constrained spaces. Moving elements should be in tune with Office. The character of add-in motion should be performant, reliable, and fluid. Instead of impeding productivity, aim to inform and direct.

Best practices

Do	Don't
Use recommended motion durations.	Don't use exaggerated animations. Avoid creating experiences that embellish and distract your customers.
Follow recommended easing curves.	Don't move elements in a jerky or disjointed manner. Avoid anticipations, bounces, rubber band, or other effects that emulate natural world physics.



See also

- Fluent UI React motion patterns
- Fabric Core animation guidelines
- Motion for Universal Windows Platform apps

Typography

Article • 08/23/2023

Segoe is the standard typeface for Office. Use it in your add-in to align with Office task panes, dialog boxes, and content objects. Fabric Core gives you access to Segoe. It provides a full type ramp of Segoe with many variations - across font weight and size - in convenient CSS classes. Not all Fabric Core sizes and weights will look great in an Office Add-in. To fit harmoniously or avoid conflicts, consider using a subset of the Fabric Core type ramp. The following table lists Fabric Core's base classes that we recommend for use in Office Add-ins.

> **Note**
>
> Text color isn't included in these base classes. Use Fabric Core's "neutral primary" for most text on white backgrounds.
>
> To learn more about available typography, see **Web Typography**.

Type	Class	Size	Weight	Recommended Usage
Hero	.ms-font-xxl	28 px	Segoe Light	• This class is larger than all other typographic elements in Office. Use it sparingly to avoid unseating visual hierarchy. • Avoid use on long strings in constrained spaces. • Provide ample whitespace around text using this class. • Commonly used for first-run messages, hero elements, or other calls to action.
Title	.ms-font-xl	21 px	Segoe Light	• This class matches the task pane title of Office applications. • Use it sparingly to avoid a flat typographic hierarchy. • Commonly used as the top-level element such as dialog box, page, or content titles.
Subtitle	.ms-font-l	17 px	Segoe Semilight	• This class is the first stop below titles. • Commonly used as a subtitle, navigation element, or group header.
Body	.ms-	14	Segoe	• Commonly used as body text within add-ins.

Type	Class	Size	Weight	Recommended Usage
	font-m	px	Regular	
Caption	.ms-font-xs	11 px	Segoe Regular	• Commonly used for secondary or tertiary text such as timestamps, by lines, captions, or field labels.
Annotation	.ms-font-mi	10 px	Segoe Semibold	• The smallest step in the type ramp should be used rarely. It's available for circumstances where legibility isn't required.

UX design patterns for Office Add-ins

Article • 02/11/2025

Designing the user experience for Office Add-ins should provide a compelling experience for Office users and extend the overall Office experience by fitting seamlessly within the default Office UI.

Our UX patterns are composed of components. Components are controls that help your customers interact with elements of your software or service. Buttons, navigation, and menus are examples of common components that often have consistent styles and behaviors.

Fluent UI React components look and behave like a part of Office, as do the framework-neutral components of Office UI Fabric JS. Take advantage of either set of components to integrate with Office. Alternatively, if your add-in has its own preexisting component language, you don't need to discard it. Look for opportunities to retain it while integrating with Office. Consider ways to swap out stylistic elements, remove conflicts, or adopt styles and behaviors that remove user confusion.

The provided patterns are best practice solutions based on common customer scenarios and user experience research. They are meant to provide both a quick entry point to designing and developing add-ins as well as guidance to achieve balance between Microsoft brand elements and your own. Providing a clean, modern user experience that balances design elements from Microsoft's Fluent UI design language and the partner's unique brand identity may help increase user retention and adoption of your add-in.

Use the UX pattern templates to:

- Apply solutions to common customer scenarios.
- Apply design best practices.
- Incorporate Fluent UI components and styles.
- Build add-ins that visually integrate with the default Office UI.
- Ideate and visualize UX.

Getting started

The patterns are organized by key actions or experiences that are common in an add-in. The main groups are:

- First-run experience (FRE)
- Authentication

- Navigation
- Branding Design

Browse each grouping to get an idea of how you can design your add-in using best practices.

> **Note**
>
> The example screens shown throughout this documentation are designed and displayed at a resolution of **1366x768**.

See also

- Office Add-in design language
- Best practices for developing Office Add-ins
- Fluent UI React in Office Add-ins