Materials

A material is a visual effect that creates a sense of depth, layering, and hierarchy between foreground and background elements.

Materials help visually separate foreground elements, such as text and controls, from background elements, such as content and solid colors. By allowing color to pass through from background to foreground, a material establishes visual hierarchy to help people more easily retain a sense of place.

Apple platforms feature two types of materials: Liquid Glass, and standard materials. Liquid Glass is a dynamic material that unifies the design language across Apple platforms, allowing you to present controls and navigation without obscuring underlying content. In contrast to Liquid Glass, the standard materials help with visual differentiation within the content layer.

Liquid Glass

Liquid Glass forms a distinct functional layer for controls and navigation elements — like tab bars and sidebars — that floats above the content layer, establishing a clear visual hierarchy between functional elements and content. Liquid Glass allows content to scroll and peek through from beneath these elements to give the interface a sense of dynamism and depth, all while maintaining legibility for controls and navigation.

Don’t use Liquid Glass in the content layer. Liquid Glass works best when it provides a clear distinction between interactive elements and content, and including it in the content layer can result in unnecessary complexity and a confusing visual hierarchy. Instead, use standard materials for elements in the content layer, such as app backgrounds. An exception to this is for controls in the content layer with a transient interactive element like sliders and toggles; in these cases, the element takes on a Liquid Glass appearance to emphasize its interactivity when a person activates it.

Use Liquid Glass effects sparingly. Standard components from system frameworks pick up the appearance and behavior of this material automatically. If you apply Liquid Glass effects to a custom control, do so sparingly. Liquid Glass seeks to bring attention to the underlying content, and overusing this material in multiple custom controls can provide a subpar user experience by distracting from that content. Limit these effects to the most important functional elements in your app. For developer guidance, see Applying Liquid Glass to custom views.

Only use clear Liquid Glass for components that appear over visually rich backgrounds. Liquid Glass provides two variants — `regular` and `clear` — that you can choose when building custom components or styling some system components.

The *regular* variant blurs and adjusts the luminosity of background content to maintain legibility of text and other foreground elements. Scroll edge effects further enhance legibility by blurring and reducing the opacity of background content. Most system components use this variant. Use the regular variant when background content might create legibility issues, or when components have a significant amount of text, such as alerts, sidebars, or popovers.

On dark background

On light background

The *clear* variant is highly translucent, which is ideal for prioritizing the visibility of the underlying content and ensuring visually rich background elements remain prominent. Use this variant for components that float above media backgrounds — such as photos and videos — to create a more immersive content experience.

For optimal contrast and legibility, determine whether to add a dimming layer behind components with clear Liquid Glass:

- If the underlying content is bright, consider adding a dark dimming layer of 35% opacity. For developer guidance, see `clear`.
- If the underlying content is sufficiently dark, or if you use standard media playback controls from AVKit that provide their own dimming layer, you don't need to apply a dimming layer.

For guidance about the use of color, see Liquid Glass color.

Standard materials

Use standard materials and effects — such as blur, vibrancy, and blending modes — to convey a sense of structure in the content beneath Liquid Glass.

Choose materials and effects based on semantic meaning and recommended usage. Avoid selecting a material or effect based on the apparent color it imparts to your interface, because system settings can change its appearance and behavior. Instead, match the material or vibrancy style to your specific use case.

Help ensure legibility by using vibrant colors on top of materials. When you use system-defined vibrant colors, you don't need to worry about colors seeming too dark, bright, saturated, or low contrast in different contexts. Regardless of the material you choose, use vibrant colors on top of it. For guidance, see System colors.

Poor contrast between the material and `systemGray3` label

Good contrast between the material and vibrant color label

Consider contrast and visual separation when choosing a material to combine with blur and vibrancy effects. For example, consider that:

- Thicker materials, which are more opaque, can provide better contrast for text and other elements with fine features.
- Thinner materials, which are more translucent, can help people retain their context by providing a visible reminder of the content that's in the background.

For developer guidance, see `Material`.

Platform considerations

iOS, iPadOS

In addition to Liquid Glass, iOS and iPadOS continue to provide four standard materials — ultra-thin, thin, regular (default), and thick — which you can use in the content layer to help create visual distinction.

iOS and iPadOS also define vibrant colors for labels, fills, and separators that are specifically designed to work with each material. Labels and fills both have several levels of vibrancy; separators have one level. The name of a level indicates the relative amount of contrast between an element and the background: The default level has the highest contrast, whereas quaternary (when it exists) has the lowest contrast.

Except for quaternary, you can use the following vibrancy values for labels on any material. In general, avoid using quaternary on top of the `thin` and `ultraThin` materials, because the contrast is too low.

- `UIVibrancyEffectStyle.label` (default)
- `UIVibrancyEffectStyle.secondaryLabel`
- `UIVibrancyEffectStyle.tertiaryLabel`
- `UIVibrancyEffectStyle.quaternaryLabel`

You can use the following vibrancy values for fills on all materials.

- `UIVibrancyEffectStyle.fill` (default)
- `UIVibrancyEffectStyle.secondaryFill`
- `UIVibrancyEffectStyle.tertiaryFill`

The system provides a single, default vibrancy value for a separator, which works well on all materials.

macOS

macOS provides several standard materials with designated purposes, and vibrant versions of all system colors. For developer guidance, see `NSVisualEffectView.Material`.

Choose when to allow vibrancy in custom views and controls. Depending on configuration and system settings, system views and controls use vibrancy to make foreground content stand out against any background. Test your interface in a variety of contexts to discover when vibrancy enhances the appearance and improves communication.

Choose a background blending mode that complements your interface design. macOS defines two modes that blend background content: behind window and within window. For developer guidance, see `NSVisualEffectView.BlendingMode`.

tvOS

In tvOS, Liquid Glass appears throughout navigation elements and system experiences such as Top Shelf and Control Center. Certain interface elements, like image views and buttons, adopt Liquid Glass when they gain focus.

In addition to Liquid Glass, tvOS continues to provide standard materials, which you can use to help define structure in the content layer. The thickness of a standard material affects how prominently the underlying content shows through. For example, consider using standard materials in the following ways:

Material	Recommended for
`ultraThin`	Full-screen views that require a light color scheme
`thin`	Overlay views that partially obscure onscreen content and require a light color scheme
`regular`	Overlay views that partially obscure onscreen content
	Overlay views that partially obscure onscreen content and require a

thick	dark color scheme

visionOS

In visionOS, windows generally use an unmodifiable system-defined material called *glass* that helps people stay grounded by letting light, the current Environment, virtual content, and objects in people's surroundings show through. Glass is an adaptive material that limits the range of background color information so a window can continue to provide contrast for app content while becoming brighter or darker depending on people's physical surroundings and other virtual content.

Note

visionOS doesn't have a distinct Dark Mode setting. Instead, glass automatically adapts to the luminance of the objects and colors behind it.

Avoid using opaque colors in a window. Areas of opacity can block people's view, making them feel constricted and reducing their awareness of the virtual and physical objects around them.

If necessary, choose materials that help you create visual separations or indicate interactivity in your app. If you need to create a custom component, you may need to specify a system material for it. Use the following examples for guidance.

- The `thin` material brings attention to interactive elements like buttons and selected items.
- The `regular` material can help you visually separate sections of your app, like a sidebar or a grouped table view.

- The `thick` material lets you create a dark element that remains visually distinct when it's on top of an area that uses a `regular` background.

To ensure foreground content remains legible when it displays on top of a material, visionOS applies vibrancy to text, symbols, and fills. Vibrancy enhances the sense of depth by pulling light and color forward from both virtual and physical surroundings.

visionOS defines three vibrancy values that help you communicate a hierarchy of text, symbols, and fills.

- Use `UIVibrancyEffectStyle.label` for standard text.
- Use `UIVibrancyEffectStyle.secondaryLabel` for descriptive text like footnotes and subtitles.
- Use `UIVibrancyEffectStyle.tertiaryLabel` for inactive elements, and only when text doesn't need high legibility.

`label`

`secondaryLabel`

`tertiaryLabel`

watchOS

Use materials to provide context in a full-screen modal view. Because full-screen modal views are common in watchOS, the contrast provided by material layers can help orient people in your app and distinguish controls and system elements from other content. Avoid removing or replacing material backgrounds for modal sheets when they're provided by default.

Resources

Related

Color

Accessibility

Dark Mode

Developer documentation

Adopting Liquid Glass

`glassEffect(_:in:)` — SwiftUI

`Material` — SwiftUI

`UIVisualEffectView` — UIKit

`NSVisualEffectView` — AppKit

Videos

Get to know the new design system

Change log

Date	Changes
September 9, 2025	Updated guidance for Liquid Glass.
June 9, 2025	Added guidance for Liquid Glass.
August 6,	Added platform-specific art.

2024	
December 5, 2023	Updated descriptions of the various material types, and clarified terms related to vibrancy and material thickness.
June 21, 2023	Updated to include guidance for visionOS.
June 5, 2023	Added guidance on using materials to provide context and orientation in watchOS apps.