

## Documentation

< All Technologies

### WidgetKit

#### WidgetRenderingMode

##### Rendering modes

- ↳ static let fullColor: WidgetRenderingMode
- ↳ static let accented: WidgetRenderingMode
- ↳ static let vibrant: WidgetRenderingMode

#### WidgetAccentedRenderingMode

#### AccessoryWidgetBackground

#### WidgetLocation

##### Timeline updates

- ↳ Keeping a widget up to date

#### TimelineProvider

#### AppIntentTimelineProvider

#### IntentTimelineProvider

#### TimelineProviderContext

#### TimelineEntry

#### Timeline

#### WidgetCenter

##### Push notification updates

≡ Filter

/

[WidgetKit](#) / [WidgetRenderingMode](#)

### Structure

## WidgetRenderingMode

Constants that indicate the rendering mode for a widget.

iOS 16.0+ | iPadOS 16.0+ | Mac Catalyst | macOS 13.0+ | visionOS 26.0+ | watchOS 9.0+

`struct WidgetRenderingMode`

## Mentioned in

- ↳ [Migrating ClockKit complications to WidgetKit](#)

## Overview

The system can modify the appearance of accessory family widgets. For example, it renders widgets on the Lock Screen on iPhone using the [vibrant](#) mode, while it renders widget-based complications in watchOS using either the [fullColor](#) or [accented](#) modes, depending on the watch face and the user's settings.

You can read the rendering mode from the environment values using the `.widgetRenderingMode` key.

`@Environment(\.widgetRenderingMode) var widgetRenderingMode`

You can then customize your widget's design based on the rendering mode.

## Topics

### Rendering modes

`static let fullColor: WidgetRenderingMode`

The system renders the widget in full color.

`static let accented: WidgetRenderingMode`

The system divides the widget's view hierarchy into an accent group and a default group, applying a different color to each group.

`static let vibrant: WidgetRenderingMode`

The system desaturates the widget, making a monochrome version that it uses to create an adaptive, vibrant effect.

## Relationships

### Conforms To

[CustomStringConvertible](#), [Equatable](#)

## See Also

### Layout and presentation

↳ [Supporting additional widget sizes](#)

Offer widgets in additional contexts by adding support for various widget sizes.

↳ [Displaying the right widget background](#)

Group your widget's background views and mark them as removable to ensure your widget appears correctly for each context and platform.

↳ [Optimizing your widget for accented rendering mode and Liquid Glass](#)

Make your widget feel at home on Apple platforms and Liquid Glass by using accented rendering mode.

↳ [Adding StandBy and CarPlay support to your widget](#)

Ensure that your small system family widget works well in StandBy and CarPlay.

`struct WidgetAccentedRenderingMode`

Constants that indicate the rendering mode for an Image in when displayed in a widget in [accented](#) mode.

`struct AccessoryWidgetBackground`

An adaptive background view that provides a standard appearance based on the the widget's environment.

`struct WidgetLocation`

Values that indicate different widget locations.

Apple > Developer > Documentation

### Platforms

iOS

iPadOS

macOS

tvOS

visionOS

watchOS

### Tools

Swift

SwiftUI

Swift Playground

TestFlight

Xcode

Xcode Cloud

SF Symbols

### Topics & Technologies

Accessibility

Accessories

App Extension

App Store

Audio & Video

Augmented Reality

Design

Distribution

Education

Fonts

Games

Health & Fitness

In-App Purchase

Localization

Maps & Location

Machine Learning & AI

Open Source

Security

Safari & Web

### Resources

Documentation

Tutorials

Downloads

Forums

Videos

Support

Support Articles

Contact Us

Bug Reporting

System Status

Feedback Assistant

### Programs

Apple Developer Program

Apple Developer Enterprise Program

App Store Small Business Program

MFi Program

News Partner Program

Video Partner Program

Security Bounty Program

Security Research Device Program

### Events

Meet with Apple

Apple Developer Centers

App Store Awards

Apple Design Awards

Apple Developer Academies

WWDC

To submit feedback on documentation, visit [Feedback Assistant](#).

Copyright © 2025 Apple Inc. All rights reserved.

[Terms of Use](#) | [Privacy Policy](#) | [Agreements and Guidelines](#)

Light Dark Auto