

Documentation

◀ All Technologies

WidgetKit

- ⓘ 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
 - ⓘ Updating widgets with WidgetKit push notifications
- > ⓘ WidgetPushHandler
- > ⓘ WidgetPushInfo

≡ Filter

/

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

Type Property

fullColor

The system renders the widget in full color.

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

`static let fullColor: WidgetRenderingMode`

Mentioned in

- ⓘ [Developing a WidgetKit strategy](#)
- ⓘ [Preparing widgets for additional platforms, contexts, and appearances](#)
- ⓘ [Updating your widgets for visionOS](#)
- ⓘ [Optimizing your widget for accented rendering mode and Liquid Glass](#)

Discussion

In this mode, the system doesn't alter or filter the widget's colors.

The system displays full-color widget-based complications on some watch faces, such as the Infograph face, on the Home Screen or Today View in iOS or iPadOS, and in Notification Center on macOS.

Note

The Infograph face only uses full-color rendering when the user sets the face to multicolor. If the user selects an accent color, the system uses [accented](#) instead.

See Also

Rendering modes

`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.

Platforms	Topics & Technologies	Resources	Programs
iOS	Accessibility	Documentation	Apple Developer Program
iPadOS	Accessories	Tutorials	Apple Developer Enterprise Program
macOS	App Extension	Downloads	App Store Small Business Program
tvOS	App Store	Forums	MFi Program
visionOS	Audio & Video	Videos	News Partner Program
watchOS	Augmented Reality		Video Partner Program
Tools	Design	Support	Security Bounty Program
Swift	Distribution	Support Articles	Security Research Device Program
SwiftUI	Education	Contact Us	Events
Swift Playground	Fonts	Bug Reporting	Meet with Apple
TestFlight	Games	System Status	Apple Developer Centers
Xcode	Health & Fitness	Apple Developer	App Store Awards
Xcode Cloud	In-App Purchase	App Store Connect	Apple Design Awards
SF Symbols	Localization	Certificates, IDs, & Profiles	Apple Developer Academies
	Maps & Location	Feedback Assistant	WWDC
	Machine Learning & AI		
	Open Source		
	Security		
	Safari & Web		