







< All Technologies





WidgetKit



-  **static let fullColor: WidgetRendering...**
-  static let accented: WidgetRenderingM...
-  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 no...

- >  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

- 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
- Account**
- Apple Developer
- App Store Connect
- Certificates, IDs, & Profiles
- 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