All Technologies

WidgetKit

- static let accented: WidgetRendering...
- 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

Capabilities

Filter /

WidgetKit / WidgetRenderingMode / accented

Type Property

accented

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

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

```
static let accented: WidgetRenderingMode
```

Mentioned in

- Developing a WidgetKit strategy
- Optimizing your widget for accented rendering mode and Liquid Glass
- Preparing widgets for additional platforms, contexts, and appearances
- Migrating ClockKit complications to WidgetKit
- Updating your widgets for visionOS

Discussion

In watchOS, the system displays accented widget-based complications on many watch faces. For example, when the user selects a color, the Infograph watch face uses white for the default group, and the user-selected color for the accent. However, these colors can change from face to face. In the X-Large watch face, the system applies the selected color to the default group, and colors the accent group white. Other faces use system-defined colors for both groups; for example, the Solar Dial face defines both an accent and a default color, and changes these colors based on the time of day.

When applying the colors, the system treats the widget’s views as if they were template images. It replaces the view’s color — rendering the new colors while preserving the view’s alpha channel.

To control your view’s appearance, add the `widgetAccentable(_:)` modifier to part of your view’s hierarchy. The system adds that view and all of its subviews to the accent group. It puts all other views in the default group.

```
var body: some View {
    VStack {
        Text("MON")
            .font(.caption)
            .widgetAccentable()
        Text("6")
            .font(.title)
    }
}
```

See Also

Rendering modes

`static let fullColor: WidgetRenderingMode`

The system renders the widget in full color.

`static let vibrant: WidgetRenderingMode`

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

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
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

To submit feedback on documentation, visit Feedback Assistant.

Light Dark Auto