Class

NSScreen

An object that describes the attributes of a computer's monitor or screen.

macOS

```
class NSScreen
```

Overview

An app may use an `NSScreen` object to retrieve information about a screen and use this information to decide what to display on that screen. For example, an app may use the `deepest` method to find out which of the available screens can best represent color and then might choose to display all of its windows on that screen.

Create the application object before you use the methods in this class, so that the application object can make the necessary connection to the window system. You can make sure the application object exists by invoking the `shared` method of `NSApplication`. If you created your app with Xcode, the application object is automatically created for you during initialization.

> Note
>
> The `NSScreen` class is only for getting information about the available displays. If you need additional information or want to change the attributes relating to a display, you must use Quartz Services. For more information, see Quartz Display Services.

Topics

Getting Screen Objects

`class var main: NSScreen?`
Returns the screen object containing the window with the keyboard focus.

`class var deepest: NSScreen?`
Returns a screen object representing the screen that can best represent color.

`class var screens: [NSScreen]`
Returns an array of screen objects representing all of the screens available on the system.

Getting Screen Information

`var depth: NSWindow.Depth`
The current bit depth and colorspace information of the screen.

`var frame: NSRect`
The dimensions and location of the screen.

`var supportedWindowDepths: UnsafePointer<NSWindow.Depth>`
A zero-terminated array of the window depths supported by the screen.

`var deviceDescription: [NSDeviceDescriptionKey : Any]`
The device dictionary for the screen.

`struct NSDeviceDescriptionKey`
These constants are the keys for device description dictionaries.

`var colorSpace: NSColorSpace?`
The color space of the screen.

`var localizedName: String`
The localized name of the display.

`func canRepresent(NSDisplayGamut) -> Bool`
A Boolean value indicating whether the color space of the screen is capable of representing the specified display gamut.

`enum NSDisplayGamut`

`class var screensHaveSeparateSpaces: Bool`
Returns a Boolean value indicating whether each screen can have its own set of spaces.

Converting Between Screen and Backing Coordinates

`func backingAlignedRect(NSRect, options: AlignmentOptions) -> NSRect`
Converts a rectangle in global screen coordinates to a pixel aligned rectangle.

`var backingScaleFactor: CGFloat`
The backing store pixel scale factor for the screen.

`func convertRectFromBacking(NSRect) -> NSRect`
Converts the rectangle from the device pixel aligned coordinates system of a screen.

`func convertRectToBacking(NSRect) -> NSRect`
Converts the rectangle to the device pixel aligned coordinates system of a screen.

Getting the Visible Portion of the Screen

`var visibleFrame: NSRect`
The current location and dimensions of the visible screen.

`var safeAreaInsets: NSEdgeInsets`
The distances from the screen's edges at which content isn't obscured.

`var auxiliaryTopLeftArea: NSRect?`
The unobscured portion of the top-left corner of the screen.

`var auxiliaryTopRightArea: NSRect?`
The unobscured portion of the top-right corner of the screen.

Getting Extended Dynamic Range Details

`var maximumPotentialExtendedDynamicRangeColorComponentValue: CGFloat`
The maximum possible color component value for the screen when it's in extended dynamic range (EDR) mode.

`var maximumExtendedDynamicRangeColorComponentValue: CGFloat`
The current maximum color component value for the screen.

`var maximumReferenceExtendedDynamicRangeColorComponentValue: CGFloat`
The current maximum color component value for reference rendering to the screen.

Getting Variable Refresh Rate Details

`var maximumFramesPerSecond: Int`
The maximum number of frames per second that the screen supports.

`var minimumRefreshInterval: TimeInterval`
The shortest refresh interval that the screen supports.

`var maximumRefreshInterval: TimeInterval`
The largest refresh interval that the screen supports.

`var displayUpdateGranularity: TimeInterval`
The number of seconds between the screen's supported update rates, for screens that support fixed update rates.

`var lastDisplayUpdateTimestamp: TimeInterval`
The time of the last framebuffer update, expressed as the number of seconds since system startup.

Receiving Screen-Related Notifications

`class let colorSpaceDidChangeNotification: NSNotification.Name`
Posted when the color space of the screen has changed.

Synchronizing with the display's refresh rate

`func displayLink(target: Any, selector: Selector) -> CADisplayLink`

Instance Properties

`var cgDirectDisplayID: CGDirectDisplayID?`
The CGDirectDisplayID for this screen. This will return nil if there isn't one and will never return kCGNullDirectDisplay.

Relationships

Inherits From

`NSObject`

Conforms To

`CVarArg`
`CustomDebugStringConvertible`
`CustomStringConvertible`
`Equatable`
`Hashable`
`NSObjectProtocol`

