

```
import Accessibility.AXAudiograph
import Accessibility.AXBrailleMap
import Accessibility.AXColorUtilities
import Accessibility.AXCustomContent
import
Accessibility.AXFeatureOverrideSessionManager
import Accessibility.AXFoundation
import Accessibility.AXHearingUtilities
import Accessibility.AXMathExpression
import Accessibility.AXRequest
import Accessibility.AXSettings
import Accessibility.AXTechnology
import Foundation
import _Concurrency
import _StringProcessing
import _SwiftConcurrencyShims
```

```
@available(iOS, introduced: 17.0,
deprecated: 18.0, renamed:
"AccessibilitySettings.animatedImagesEnabled")
```

```
@available(macOS, introduced: 14.0,
deprecated: 15.0, renamed:
"AccessibilitySettings.animatedImagesEnabled")
```

```
@available(tvOS, introduced: 17.0,
deprecated: 18.0, renamed:
"AccessibilitySettings.animatedImagesEnabled")
```

```
@available(watchOS, introduced: 10.0,
deprecated: 11.0, renamed:
"AccessibilitySettings.animatedImagesEnabled")
```

```

led")
public func AXAnimatedImagesEnabled() ->
Bool

@available(iOS, introduced: 17.0,
deprecated: 18.0, renamed:
"AccessibilitySettings.prefersHorizontalT
extLayout")
@available(macOS, introduced: 14.0,
deprecated: 15.0, renamed:
"AccessibilitySettings.prefersHorizontalT
extLayout")
@available(tvOS, introduced: 17.0,
deprecated: 18.0, renamed:
"AccessibilitySettings.prefersHorizontalT
extLayout")
@available(watchOS, introduced: 10.0,
deprecated: 11.0, renamed:
"AccessibilitySettings.prefersHorizontalT
extLayout")
public func
AXPrefersHorizontalTextLayout() -> Bool

/// An accessibility notification that an
app can send.
@available(iOS 17.0, macOS 14.0, tvOS
17.0, watchOS 10.0, *)
public enum AccessibilityNotification {

    /// A notification that an app posts
when it needs to convey an
    /// announcement to the assistive
app. Include an announcement value,

```

```
    /// such as `String` or  
    `AttributedString`, that an assistive app  
    will
```

```
    /// announce.  
    public struct Announcement {  
  
        public init(_ announcement:  
AttributedString)  
  
        public init(_ announcement:  
String)  
  
        public init(_ announcement:  
NSAttributedString)  
    }
```

```
    /// A notification that an app posts  
    when the layout of a screen changes.
```

```
    /// Optionally, include a parameter  
    that contains the accessibility element  
    /// for VoiceOver to move to after  
    processing the notification.
```

```
    public struct LayoutChanged {  
  
        public init(_ element: Any? =  
nil)  
    }
```

```
    /// A notification that an app posts  
    when a new view appears that occupies
```

```
    /// a major portion of the screen.  
    Optionally, include a parameter that  
    /// contains the accessibility
```

```

element for VoiceOver to move to after
    /// processing the notification.
    public struct ScreenChanged {

        public init(_ element: Any? =
nil)
    }

    /// A notification that an app posts
when a scroll action completes.
    /// Include an announcement value,
such as `String` or `AttributedString`,
    /// that an assistive app will
announce. When an assistive app
repeatedly
    /// receives the same scroll position
value, it indicates to users that
    /// scrolling can't continue due to a
border or boundary.
    public struct PageScrolled {

        public init(_ announcement:
AttributedString)

        public init(_ announcement:
String)

        public init(_ announcement:
NSAttributedString)
    }
}

@available(macOS 12.2, iOS 15.2, tvOS

```

```

15.2, watchOS 8.2, *)
extension AXBrailleMap {

    public subscript(point: CGPoint) ->
Float
}

@available(macOS 12.0, iOS 15.0, tvOS
15.0, watchOS 8.0, *)
extension AXNumericDataAxisDescriptor {

    public var range: ClosedRange<Double>

    public var gridlinePositions:
[Double]

    public convenience init(title:
String, range: ClosedRange<Double>,
gridlinePositions: [Double],
valueDescriptionProvider: @escaping
((Double) -> String))

    public convenience
init(attributedTitle: NSAttributedString,
range: ClosedRange<Double>,
gridlinePositions: [Double],
valueDescriptionProvider: @escaping
((Double) -> String))
}

@available(macOS 12.0, iOS 15.0, tvOS
15.0, watchOS 8.0, *)
extension AXDataPoint {

```

```

    public enum Value {

        case number(Double)

        case category(String)
    }

    /// Initializer for a data point with
    a numeric X value.
    public convenience init(x: Double, y:
    Double? = nil, additionalValues:
    [AXDataPoint.Value] = [], label: String?
    = nil)

    /// Initializer for a data point with
    a categorical X value.
    public convenience init(x: String, y:
    Double? = nil, additionalValues:
    [AXDataPoint.Value] = [], label: String?
    = nil)
    }

    @available(macOS 12.0, iOS 15.0, tvOS
    15.0, watchOS 8.0, *)
    extension AXChartDescriptor {

        public var xAxis: any
        AXDataAxisDescriptor

        public var additionalAxes: [any
        AXDataAxisDescriptor]
    }

```

```
    public convenience init(title:
String? = nil, summary: String? = nil,
xAxis: any AXDataAxisDescriptor, yAxis:
AXNumericDataAxisDescriptor? = nil,
additionalAxes: [any
AXDataAxisDescriptor] = [], series:
[AXDataSeriesDescriptor])
```

```
    public convenience
init(attributedTitle: NSAttributedString?
= nil, summary: String? = nil, xAxis: any
AXDataAxisDescriptor, yAxis:
AXNumericDataAxisDescriptor? = nil,
additionalAxes: [any
AXDataAxisDescriptor] = [], series:
[AXDataSeriesDescriptor])
}
```

```
@available(macOS 12.0, iOS 15.0, tvOS
15.0, watchOS 8.0, *)
extension AttributeScopes {
```

```
    /// The scope of accessibility
related attributes
    public var accessibility:
AttributeScopes.AccessibilityAttributes.T
ype { get }
```

```
    /// A scope of accessibility
attributes to enrich how assistive
technologies
    /// will describe text.
    public struct AccessibilityAttributes
```

```
: AttributeScope {  
  
    /// The level of this heading.  
    public let  
accessibilityHeadingLevel:  
AttributeScopes.AccessibilityAttributes.H  
eadingLevelAttribute  
  
    /// Custom, localized text  
attributes.  
    public let  
accessibilityTextCustom:  
AttributeScopes.AccessibilityAttributes.T  
extCustomAttribute  
  
    /// The textual context.  
    public let  
accessibilityTextualContext:  
AttributeScopes.AccessibilityAttributes.T  
extualContextAttribute  
  
    /// Customizes whether  
punctuation should be spoken.  
    public let  
accessibilitySpeechIncludesPunctuation:  
AttributeScopes.AccessibilityAttributes.I  
ncludesPunctuationAttribute  
  
    /// Customizes the pitch the  
spoken speech.  
    public let  
accessibilitySpeechAdjustedPitch:  
AttributeScopes.AccessibilityAttributes.A
```


djustedPitchAttribute

```
    /// Customizes whether each
    character should be spoken separately.
    public let
    accessibilitySpeechSpellsOutCharacters:
    AttributeScopes.AccessibilityAttributes.S
    pellOutAttribute

    /// Customizes if speech
    announcements should be queued.
    @available(*, deprecated,
    renamed:
    "accessibilitySpeechAnnouncementPriority"
    )
    public let
    accessibilitySpeechAnnouncementsQueued:
    AttributeScopes.AccessibilityAttributes.Q
    ueueAnnouncementAttribute

    /// The International Phonetic
    Alphabet representation for speech.
    public let
    accessibilitySpeechPhoneticNotation:
    AttributeScopes.AccessibilityAttributes.I
    PAnotationAttribute

    /// Customizes whether speech
    announcements should be queued or
    interrupted.
    @available(iOS 17.0, macOS 14.0,
    tvOS 17.0, watchOS 10.0, *)
    public let
```

```
accessibilitySpeechAnnouncementPriority:
AttributeScopes.AccessibilityAttributes.A
nnouncementPriorityAttribute
```

```
        @available(iOS 15.0, tvOS 15.0,
watchOS 8.0, macOS 12.0, *)
        public typealias
```

```
DecodingConfiguration =
AttributeScopeCodableConfiguration
```

```
        @available(iOS 15.0, tvOS 15.0,
watchOS 8.0, macOS 12.0, *)
        public typealias
```

```
EncodingConfiguration =
AttributeScopeCodableConfiguration
```

```
    }
}
```

```
@available(macOS 12.0, iOS 15.0, tvOS
15.0, watchOS 8.0, *)
extension AttributeDynamicLookup {
```

```
    public subscript<T>(dynamicMember
keyPath:
KeyPath<AttributeScopes.AccessibilityAttr
ibutes, T>) -> T where T :
AttributedStringKey { get }
}
```

```
extension AccessibilitySettings {
```

```
    /// Returns whether the system
setting to prefer non-blinking text
```

insertion indicators in editable text fields is enabled. Apps that draw custom insertion indicators should honor this setting.

```
@available(macOS 15.0, iOS 18.0, tvOS 18.0, watchOS 11.0, visionOS 2.0, *)
public static var
prefersNonBlinkingTextInsertionIndicator:
Bool { get }
}
```

```
extension AccessibilitySettings {
```

```
    /// Returns whether the system
    setting to prefer horizontal text is
    enabled for languages that support both
    vertical and horizontal text layout.
```

```
    @available(macOS 15.0, iOS 18.0, tvOS 18.0, watchOS 11.0, visionOS 2.0, *)
    public static var
    prefersHorizontalTextLayout: Bool { get }
```

```
    @available(macOS 15.0, iOS 18.0, tvOS 18.0, watchOS 11.0, visionOS 2.0, *)
    public static var
    prefersHorizontalTextLayoutDidChangeNotif
ication: Notification.Name { get }
}
```

```
extension AccessibilitySettings {
```

```
    /// Returns whether the system
    setting for displaying animations in an
```

animated image is on or off.

```
@available(macOS 15.0, iOS 18.0, tvOS
18.0, watchOS 11.0, visionOS 2.0, *)
public static var
animatedImagesEnabled: Bool { get }

    @available(macOS 15.0, iOS 18.0, tvOS
18.0, watchOS 11.0, visionOS 2.0, *)
    public static var
animatedImagesEnabledDidChangeNotificatio
n: Notification.Name { get }
}
```

```
extension AccessibilitySettings {
```

```
    @available(macOS 15.0, iOS 18.0, tvOS
18.0, watchOS 11.0, visionOS 2.0, *)
    public static func openSettings(for
feature: AccessibilitySettings.Feature)
async throws
}
```

```
extension NSNotification.Name {
```

```
    @available(iOS, introduced: 17.0,
deprecated: 18.0, renamed:
"AccessibilitySettings.prefersHorizontalT
extLayout")
    @available(macOS, introduced: 14.0,
deprecated: 15.0, renamed:
"AccessibilitySettings.prefersHorizontalT
extLayoutDidChangeNotification")
    @available(tvOS, introduced: 17.0,
```

```
deprecated: 18.0, renamed:
"AccessibilitySettings.prefersHorizontalT
extLayoutDidChangeNotification")
    @available(watchOS, introduced: 10.0,
deprecated: 11.0, renamed:
"AccessibilitySettings.prefersHorizontalT
extLayoutDidChangeNotification")
    public static let
AXPrefersHorizontalTextLayoutDidChange:
Notification.Name

    @available(iOS, introduced: 17.0,
deprecated: 18.0, renamed:
"AccessibilitySettings.animatedImagesEnab
ledDidChangeNotification")
    @available(macOS, introduced: 14.0,
deprecated: 15.0, renamed:
"AccessibilitySettings.animatedImagesEnab
ledDidChangeNotification")
    @available(tvOS, introduced: 17.0,
deprecated: 18.0, renamed:
"AccessibilitySettings.animatedImagesEnab
ledDidChangeNotification")
    @available(watchOS, introduced: 10.0,
deprecated: 11.0, renamed:
"AccessibilitySettings.animatedImagesEnab
ledDidChangeNotification")
    public static let
AXAnimatedImagesEnabledDidChange:
Notification.Name
}
```