NSRulerMarker Class Objective-C Reference

Cocoa > Text & Fonts

2006-05-23

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

NSRulerMarker Class Objective-C Reference

Inherits from	NSObject
Conforms to	NSCoding NSCopying NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.0 and later.
Companion guide	Rulers and Paragraph Styles
Declared in	NSRulerMarker.h
Related sample code	Sketch-112

Overview

An NSRulerMarker object displays a symbol on an NSRulerView object, indicating a location for whatever graphics element it represents in the client of the ruler view (for example, a margin or tab setting, or the edges of a graphic on the page).

Adopted Protocols

NSCoding
- `encodeWithCoder:`
- `initWithCoder:`

NSCopying
- `copyWithZone:`

Tasks

Creating Instances

`– initWithRulerView:markerLocation:image:imageOrigin:` (page 9)
Initializes a newly allocated ruler marker, associating it with (but not adding it to) a specified ruler view and assigning the attributes provided.

Getting the Ruler View

`– ruler` (page 12)
Returns the receiver's ruler view.

Setting the Image

`– setImage:` (page 12)
Sets the receiver’s image.

`– image` (page 8)
Returns the image displayed by the receiver.

`– setImageOrigin:` (page 12)
Sets the point in the receiver’s image that is positioned at the receiver’s location on the ruler view.

`– imageOrigin` (page 8)
Returns the point in the receiver’s image positioned at the receiver’s location on the ruler view.

`– imageRectInRuler` (page 8)
Returns the rectangle occupied by the receiver’s image.

`– thicknessRequiredInRuler` (page 15)
Returns the amount of the receiver’s image that’s displayed above or to the left of the ruler view's baseline.

Setting Movability

`– setMovable:` (page 13)
Sets whether the user can move the receiver in its ruler view.

`– isMovable` (page 10)
Returns whether the user can move the receiver on its ruler view.

`– setRemovable:` (page 14)
Sets whether the user can remove the receiver from its ruler view.

`– isRemovable` (page 10)
Returns whether the user can remove the receiver from its ruler view.

Setting the Location

- `setMarkerLocation:` (page 13)
  Sets the location of the receiver in the coordinate system of the ruler view's client view.
- `markerLocation` (page 11)
  Returns the location of the receiver in the coordinate system of the ruler view's client view.

Setting the Represented Object

- `setRepresentedObject:` (page 14)
  Sets the object the receiver represents.
- `representedObject` (page 11)
  Returns the object the receiver represents.

Drawing and Event Handling

- `drawRect:` (page 7)
  Draws the receiver's image that appears in the supplied rectangle.
- `isDragging` (page 10)
  Returns whether the receiver is being dragged.
- `trackMouse:adding:` (page 15)
  Handles user manipulation of the receiver in its ruler view.

Instance Methods

drawRect:

Draws the receiver's image that appears in the supplied rectangle.

```
- (void)drawRect:(NSRect)aRect
```

Parameters

aRect

The rectangle to be drawn, in the ruler view's coordinate system.

Availability

Available in Mac OS X v10.0 and later.

See Also

- `imageRectInRuler` (page 8)

Declared In

`NSRulerMarker.h`

image

Returns the image displayed by the receiver.

```
- (NSImage *)image
```

Return Value
The image displayed by the receiver.

Availability
Available in Mac OS X v10.0 and later.

See Also
`- setImage:` (page 12)

Declared In
`NSRulerMarker.h`

imageOrigin

Returns the point in the receiver's image positioned at the receiver's location on the ruler view.

```
- (NSPoint)imageOrigin
```

Return Value
The point in the receiver's image positioned at the receiver's location on the ruler view, expressed in the image's coordinate system.

Discussion
For a horizontal ruler, the x coordinate of the image origin is aligned with the location of the marker, and the y coordinate lies on the baseline of the ruler. For vertical rulers, the y coordinate of the image origin is the location, and the x coordinate lies on the baseline.

Availability
Available in Mac OS X v10.0 and later.

See Also
`- setImageOrigin:` (page 12)
`- imageRectInRuler` (page 8)

Declared In
`NSRulerMarker.h`

imageRectInRuler

Returns the rectangle occupied by the receiver's image.

```
- (NSRect)imageRectInRuler
```

Return Value
The rectangle occupied by the receiver's image, in the ruler view's coordinate system, accounting for whether the ruler view's coordinate system is flipped.

Availability
Available in Mac OS X v10.0 and later.

See Also
- `drawRect:` (page 7)
- `thicknessRequiredInRuler` (page 15)

Declared In
`NSRulerMarker.h`

initWithRulerView:markerLocation:image:imageOrigin:

Initializes a newly allocated ruler marker, associating it with (but not adding it to) a specified ruler view and assigning the attributes provided.

```
- (id)initWithRulerView:(NSRulerView *)aRulerView markerLocation:(CGFloat)location
     image:(NSImage *)anImage imageOrigin:(NSPoint)imageOrigin
```

Parameters

aRulerView
The ruler view with which to associate the ruler marker. This method raises an `NSInvalidArgumentException` if *aRulerView* is `nil`.

location
The x or y position of the marker in the client view's coordinate system, depending on whether the ruler view is horizontal or vertical.

anImage
The image displayed at the marker location. This method raises an `NSInvalidArgumentException` if *anImage* is `nil`.

imageOrigin
The point within the image positioned at the marker location, expressed in pixels relative to the lower-left corner of the image.

Return Value
An initialized ruler marker object.

Discussion
The image used to draw the marker must be appropriate for the orientation of the ruler. Markers may need to look different on a horizontal ruler than on a vertical ruler, and the ruler view neither scales nor rotates the images.

To add the new ruler marker to *aRulerView*, use either of NSRulerView's `addMarker:` or `trackMarker:withMouseEvent:` methods. `addMarker:` immediately puts the marker on the ruler, while `trackMarker:withMouseEvent:` allows the client view to intercede in the addition and placement of the marker.

A new ruler marker can be moved on its ruler view, but not removed. Use `setMovable:` (page 13) and `setRemovable:` (page 14) to change these attributes. The new ruler marker also has no represented object; use `setRepresentedObject:` (page 14) to set one.

This method is the designated initializer for the NSRulerMarker class.

Availability
Available in Mac OS X v10.0 and later.

See Also

- `setMarkerLocation:` (page 13)
- `setImage:` (page 12)
- `setImageOrigin:` (page 12)

Declared In

`NSRulerMarker.h`

isDragging

Returns whether the receiver is being dragged.

```
- (BOOL)isDragging
```

Return Value

`YES` if the receiver is being dragged, `NO` otherwise.

Availability

Available in Mac OS X v10.0 and later.

See Also

- `trackMouse:adding:` (page 15)

Declared In

`NSRulerMarker.h`

isMovable

Returns whether the user can move the receiver on its ruler view.

```
- (BOOL)isMovable
```

Return Value

`YES` if the user can move the receiver on its ruler view, `NO` otherwise.

Discussion

By default ruler markers are movable.

Availability

Available in Mac OS X v10.0 and later.

See Also

- `setMovable:` (page 13)
- `isRemovable` (page 10)

Declared In

`NSRulerMarker.h`

isRemovable

Returns whether the user can remove the receiver from its ruler view.

```
- (BOOL)isRemovable
```

Return Value

`YES` if the user can remove the receiver from its ruler view, `NO` otherwise.

Discussion

By default ruler markers are not removable.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- setRemovable:` (page 14)

`- isMovable` (page 10)

Declared In

`NSRulerMarker.h`

markerLocation

Returns the location of the receiver in the coordinate system of the ruler view's client view.

```
- (CGFloat)markerLocation
```

Return Value

The location of the receiver in the coordinate system of the ruler view's client view. This is an x position for a horizontal ruler, a y position for a vertical ruler.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- setMarkerLocation:` (page 13)

Declared In

`NSRulerMarker.h`

representedObject

Returns the object the receiver represents.

```
- (id < NSCopying >)representedObject
```

Return Value

The object the receiver represents.

Discussion

See About Ruler Markers for more information on the represented object.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- setRepresentedObject:` (page 14)

Declared In

`NSRulerMarker.h`

ruler

Returns the receiver's ruler view.

```
- (NSRulerView *)ruler
```

Return Value

The receiver's ruler view.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- addMarker:` (NSRulerView)

Declared In

`NSRulerMarker.h`

setImage:

Sets the receiver’s image.

```
- (void)setImage:(NSImage *)anImage
```

Parameters

anImage

The new image.

Discussion

The image used to draw the marker must be appropriate for the orientation of the ruler. Markers may need to look different on a horizontal ruler than on a vertical ruler, and the ruler view neither scales nor rotates the images.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- image` (page 8)

`- setImageOrigin:` (page 12)

Declared In

`NSRulerMarker.h`

setImageOrigin:

Sets the point in the receiver’s image that is positioned at the receiver’s location on the ruler view.

```
- (void)setImageOrigin:(NSPoint)aPoint
```

Parameters

`aPoint`

The point within the image positioned at the marker location, expressed in pixels relative to the lower-left corner of the image.

Discussion

For a horizontal ruler, the x coordinate of the image origin is aligned with the location of the marker, and the y coordinate lies on the baseline of the ruler. For vertical rulers, the y coordinate of the image origin is the location, and the x coordinate lies on the baseline.

Availability

Available in Mac OS X v10.0 and later.

See Also

`– imageOrigin` (page 8)

`– setImage:` (page 12)

`– setMarkerLocation:` (page 13)

Declared In

`NSRulerMarker.h`

setMarkerLocation:

Sets the location of the receiver in the coordinate system of the ruler view's client view.

```
- (void)setMarkerLocation:(CGFloat)location
```

Parameters

`location`

The location of the receiver in the coordinate system of the ruler view's client view. This is an x position for a horizontal ruler, a y position for a vertical ruler.

Availability

Available in Mac OS X v10.0 and later.

See Also

`– markerLocation` (page 11)

`– setImageOrigin:` (page 12)

Declared In

`NSRulerMarker.h`

setMovable:

Sets whether the user can move the receiver in its ruler view.

```
- (void)setMovable:(BOOL)flag
```

Parameters

`flag`

`YES` to allow the user to drag the marker image in the ruler, `NO` to make it immobile.

Discussion
By default ruler markers are movable.

Availability
Available in Mac OS X v10.0 and later.

See Also
- `isMovable` (page 10)
- `setRemovable:` (page 14)

Declared In
`NSRulerMarker.h`

setRemovable:

Sets whether the user can remove the receiver from its ruler view.

```
- (void)setRemovable:(BOOL)flag
```

Parameters

flag
YES to allow the user to drag the marker image off of the ruler and remove the marker, NO to prevent the user from removing the marker.

Discussion
By default ruler markers are not removable.

Availability
Available in Mac OS X v10.0 and later.

See Also
- `isRemovable` (page 10)
- `setMovable:` (page 13)

Declared In
`NSRulerMarker.h`

setRepresentedObject:

Sets the object the receiver represents.

```
- (void)setRepresentedObject:(id < NSCopying >)anObject
```

Parameters

anObject
The new represented object.

Discussion
See About Ruler Markers for more information on the represented object.

Availability
Available in Mac OS X v10.0 and later.

See Also

- `representedObject` (page 11)

Declared In

`NSRulerMarker.h`

thicknessRequiredInRuler

Returns the amount of the receiver's image that's displayed above or to the left of the ruler view's baseline.

```
- (CGFloat)thicknessRequiredInRuler
```

Return Value

The amount of the receiver's image that's displayed above or to the left of the ruler view's baseline, the height for a horizontal ruler or width for a vertical ruler.

Availability

Available in Mac OS X v10.0 and later.

See Also

- `imageOrigin` (page 8)

Declared In

`NSRulerMarker.h`

trackMouse:adding:

Handles user manipulation of the receiver in its ruler view.

```
- (BOOL)trackMouse:(NSEvent *)theEvent adding:(BOOL)flag
```

Parameters

theEvent

The event that represents the user manipulation being attempted on the ruler marker.

flag

`YES` to indicate that the receiver is a new marker being added to its ruler view, `NO` otherwise.

Return Value

`YES` if the user manipulation was allowed, `NO` if it was not allowed.

Discussion

`NSRulerView` objects invoke this method automatically to add a new marker or to move or remove an existing marker. You should never need to invoke it directly.

If the receiver is a new marker being added to its ruler view (*flag* is `YES`), the receiver queries the ruler view's client before adding itself to the ruler view. If the client responds to `rulerView:shouldAddMarker:` and that method returns `NO`, this method immediately returns `NO`, and the new marker isn't added.

If the receiver is not a new marker being added to its ruler view (*flag* is `NO`), this method attempts to move or remove an existing marker, once again based on responses from the ruler view's client view. If the receiver is neither movable nor removable, this method immediately returns `NO`. Further, if the ruler view's client responds to `rulerView:shouldMoveMarker:` and returns `NO`, this method returns `NO`, indicating the receiver can't be moved.

If the receiver is being added or moved, this method queries the client view using `rulerView:willAddMarker:atLocation:` or `rulerView:willMoveMarker:toLocation:`, respectively. If the client responds to the method, the return value is used as the receiver's location. These methods are invoked repeatedly as the receiver is dragged within the ruler view.

If the receiver is an existing marker being removed (dragged off the ruler), this method queries the client view using `rulerView:shouldRemoveMarker:`. If the client responds to this method and returns `NO`, the marker is pinned to the ruler view's baseline (following the cursor on the baseline if it's movable).

When the user releases the mouse button, this method informs the client view of the marker's new status using `rulerView:didAddMarker:`, `rulerView:didMoveMarker:`, or `rulerView:didRemoveMarker:` as appropriate. The client view can use this notification to set the marker's represented object, modify its state and redisplay (for example, adjusting text layout around a new tab stop), or take whatever other action it might need.

If *`flag`* is `YES` and the user dragged the new marker away from the ruler, the marker isn't added, no message is sent, and this method returns `NO`.

See *Rulers and Paragraph Styles* for more information on these client methods.

Availability
Available in Mac OS X v10.0 and later.

See Also
`– isMovable` (page 10)
`– isRemovable` (page 10)

Declared In
`NSRulerMarker.h`

Document Revision History

This table describes the changes to *NSRulerMarker Class Objective-C Reference*.

Date	Notes
2006-05-23	First publication of this content as a separate document.

Index