NSTextContainer Class 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

NSTextContainer Class Reference

Inherits from	NSObject
Conforms to	NSCoding NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.0 and later.
Companion guide	Text System Overview
Declared in	NSTextContainer.h
Related sample code	Quartz Composer WWDC 2005 TextEdit Sketch-112 TextEditPlus TextLayoutDemo Worm

Overview

An `NSTextContainer` object defines a region where text is laid out. An NSLayoutManager uses NSTextContainers to determine where to break lines, lay out portions of text, and so on. NSTextContainer defines rectangular regions, but you can create subclasses that define regions of other shapes, such as circular regions, regions with holes in them, or regions that flow alongside graphics.

Adopted Protocols

NSCoding

- `– encodeWithCoder:`
- `– initWithCoder:`

Tasks

Creating an Instance

- `initWithContainerSize:` (page 8)
  Initializes a text container with a specified bounding rectangle.

Managing Text Components

- `setLayoutManager:` (page 13)
  Sets the receiver's layout manager.
- `layoutManager` (page 10)
  Returns the receiver's layout manager.
- `replaceLayoutManager:` (page 11)
  Replaces the layout manager for the group of text system objects containing the receiver.
- `setTextView:` (page 14)
  Sets the receiver's text view.
- `textView` (page 15)
  Returns the receiver's text view.

Controlling Size

- `setContainerSize:` (page 12)
  Sets the size of the receiver's bounding rectangle.
- `containerSize` (page 7)
  Returns the size of the receiver's bounding rectangle, regardless of the size of its region.
- `setWidthTracksTextView:` (page 15)
  Controls whether the receiver adjusts the width of its bounding rectangle when its text view is resized.
- `widthTracksTextView` (page 16)
  Returns whether the receiver adjusts the width of its bounding rectangle when its text view is resized.
- `setHeightTracksTextView:` (page 12)
  Controls whether the receiver adjusts the height of its bounding rectangle when its text view is resized.
- `heightTracksTextView` (page 8)
  Returns whether the receiver adjusts the height of its bounding rectangle when its text view is resized.

Setting Line Fragment Padding

- `setLineFragmentPadding:` (page 13)
  Sets the amount by which text is inset within line fragment rectangles.
- `lineFragmentPadding` (page 10)
  Returns the amount by which text is inset within line fragment rectangles.

Calculating Text Layout

`- lineFragmentRectForProposedRect:sweepDirection:movementDirection:remainingRect:` (page 10)

Overridden by subclasses to calculate and return the longest rectangle available in the proposed rectangle for displaying text, or `NSZeroRect` if there is none according to the receiver's region definition.

`- isSimpleRectangularTextContainer` (page 9)

Overridden by subclasses to return whether the receiver's region is a rectangle with no holes or gaps and whose edges are parallel to the text view's coordinate system axes.

Mouse Hit Testing

`- containsPoint:` (page 7)

Overridden by subclasses to return whether a point lies within the receiver's region or on the region's edge—not simply within its bounding rectangle.

Instance Methods

containerSize

Returns the size of the receiver's bounding rectangle, regardless of the size of its region.

```
- (NSSize)containerSize
```

Return Value
The size of the receiver's bounding rectangle, regardless of the size of its region.

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

See Also
`- textContainerInset` (NSTextView)
`- setContainerSize:` (page 12)

Related Sample Code
TextLayoutDemo

Declared In
`NSTextContainer.h`

containsPoint:

Overridden by subclasses to return whether a point lies within the receiver's region or on the region's edge—not simply within its bounding rectangle.

```
- (BOOL)containsPoint:(NSPoint)aPoint
```

Parameters

aPoint

The point in question.

Return Value

YES if *aPoint* lies within the receiver's region or on the region's edge—not simply within its bounding rectangle—NO otherwise.

Discussion

For example, if the receiver defines a donut shape and *aPoint* lies in the hole, this method returns NO. This method can be used for hit testing of mouse events.

NSTextContainer's implementation merely checks that *aPoint* lies within its bounding rectangle.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSTextContainer.h

heightTracksTextView

Returns whether the receiver adjusts the height of its bounding rectangle when its text view is resized.

```
- (BOOL)heightTracksTextView
```

Return Value

YES if the receiver adjusts the height of its bounding rectangle when its text view is resized, NO otherwise.

Discussion

If the receiver does track the text view height, its height is adjusted to the height of the text view minus twice the inset height (as given by NSTextView's textContainerInset method).

See *Text System Storage Layer Overview* for more information on size tracking.

Availability

Available in Mac OS X v10.0 and later.

See Also

- widthTracksTextView (page 16)
- setHeightTracksTextView: (page 12)

Declared In

NSTextContainer.h

initWithContainerSize:

Initializes a text container with a specified bounding rectangle.

```
- (id)initWithContainerSize:(NSSize)aSize
```

Parameters

aSize

The size of the text container's bounding rectangle.

Return Value

The newly initialized text container.

Discussion

The new text container must be added to an `NSLayoutManager` object before it can be used. The text container must also have an `NSTextView` object set for text to be displayed. This method is the designated initializer for the `NSTextContainer` class.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- addTextContainer:` (NSLayoutManager)

`- setTextView:` (page 14)

Related Sample Code

Quartz Composer WWDC 2005 TextEdit

Sketch-112

TextEditPlus

TextViewConfig

Declared In

`NSTextContainer.h`

isSimpleRectangularTextContainer

Overridden by subclasses to return whether the receiver's region is a rectangle with no holes or gaps and whose edges are parallel to the text view's coordinate system axes.

```
- (BOOL)isSimpleRectangularTextContainer
```

Return Value

`YES` if the receiver's region is a rectangle with no holes or gaps and whose edges are parallel to the text view's coordinate system axes, `NO` otherwise.

Discussion

A text container whose shape changes can return `YES` if its region is currently a simple rectangle, but when its shape does change it must send `textContainerChangedGeometry:` to its layout manager so the layout can be recalculated.

`NSTextContainer`'s implementation of this method returns `YES`.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

TextLayoutDemo

Declared In

`NSTextContainer.h`

layoutManager

Returns the receiver's layout manager.

```
- (NSLayoutManager *)layoutManager
```

Return Value

The text container's layout manager.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- setLayoutManager:` (page 13)

`- replaceLayoutManager:` (page 11)

Related Sample Code

Quartz Composer WWDC 2005 TextEdit

TextEditPlus

Declared In

`NSTextContainer.h`

lineFragmentPadding

Returns the amount by which text is inset within line fragment rectangles.

```
- (CGFloat)lineFragmentPadding
```

Return Value

The amount by which text is inset within line fragment rectangles, in points.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- lineFragmentRectForProposedRect:sweepDirection:movementDirection:remainingRect:` (page 10)

`- setLineFragmentPadding:` (page 13)

Related Sample Code

Quartz Composer WWDC 2005 TextEdit

TextEditPlus

Declared In

`NSTextContainer.h`

lineFragmentRectForProposedRect:sweepDirection:movementDirection:remainingRect:

Overridden by subclasses to calculate and return the longest rectangle available in the proposed rectangle for displaying text, or `NSZeroRect` if there is none according to the receiver's region definition.

```
- (NSRect)lineFragmentRectForProposedRect:(NSRect)proposedRect
    sweepDirection:(NSLineSweepDirection)sweepDirection
    movementDirection:(NSLineMovementDirection)movementDirection
    remainingRect:(NSRectPointer)remainingRect
```

Parameters

proposedRect

The proposed rectangle in which to layout text.

sweepDirection

The line sweep direction.

movementDirection

The line movement direction.

remainingRect

Upon return, the unused, possibly shifted, portion of *proposedRect* that's available for further text, or NSZeroRect if there is no remainder.

Return Value

The longest rectangle available in the proposed rectangle for displaying text, or NSZeroRect if there is none according to the receiver's region definition.

Discussion

There is no guarantee as to the width of the proposed rectangle or to its location. For example, the proposed rectangle is likely to be much wider than the width of the receiver. The receiver should examine *proposedRect* to see that it intersects its bounding rectangle and should return a modified rectangle based on *sweepDirection* and *movementDirection*, whose possible values are listed in the class description. If *sweepDirection* is NSLineSweepRight, for example, the receiver uses this information to trim the right end of *proposedRect* as needed rather than the left end.

If *proposedRect* doesn't completely overlap the region along the axis of *movementDirection* and *movementDirection* isn't NSLineDoesntMove, this method can either shift the rectangle in that direction as much as needed so that it does completely overlap, or return NSZeroRect to indicate that the proposed rectangle simply doesn't fit.

See the class description for more information on overriding this method.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSTextContainer.h

replaceLayoutManager:

Replaces the layout manager for the group of text system objects containing the receiver.

```
- (void)replaceLayoutManager:(NSLayoutManager *)aLayoutManager
```

Parameters

aLayoutManager

The new layout manager.

Discussion

All text containers and text views sharing the original layout manager share the new layout manager. This method makes all the adjustments necessary to keep these relationships intact, unlike `setLayoutManager:` (page 13).

Availability

Available in Mac OS X v10.0 and later.

See Also

`- layoutManager` (page 10)

Declared In

`NSTextContainer.h`

setContainerSize:

Sets the size of the receiver's bounding rectangle.

```
- (void)setContainerSize:(NSSize)aSize
```

Parameters

aSize

The new size of the text container's bounding rectangle.

Discussion

This method also sends `textContainerChangedGeometry:` to the text container's layout manager.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- setTextContainerInset:` (NSTextView)

`- containerSize` (page 7)

Related Sample Code

Quartz Composer WWDC 2005 TextEdit

Sketch-112

TextEditPlus

TextLayoutDemo

Declared In

`NSTextContainer.h`

setHeightTracksTextView:

Controls whether the receiver adjusts the height of its bounding rectangle when its text view is resized.

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

Parameters

flag

`YES` if the receiver should follow changes to the height of its text view, `NO` otherwise.

Discussion

See *Text System Storage Layer Overview* for more information on size tracking.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- setContainerSize:` (page 12)

`- setWidthTracksTextView:` (page 15)

`- heightTracksTextView` (page 8)

Related Sample Code

Quartz Composer WWDC 2005 TextEdit

Sketch-112

TextEditPlus

TextLayoutDemo

Declared In

`NSTextContainer.h`

setLayoutManager:

Sets the receiver's layout manager.

```
- (void)setLayoutManager:(NSLayoutManager *)aLayoutManager
```

Parameters

aLayoutManager

The new layout manager.

Discussion

This method is invoked automatically when you add a text container to a layout manager; you should never need to invoke it directly, but might want to override it. If you want to replace the layout manager for an established group of text system objects, use `replaceLayoutManager:` (page 11).

Availability

Available in Mac OS X v10.0 and later.

See Also

`- addTextContainer:` (NSLayoutManager)

`- layoutManager` (page 10)

Declared In

`NSTextContainer.h`

setLineFragmentPadding:

Sets the amount by which text is inset within line fragment rectangles.

```
- (void)setLineFragmentPadding:(CGFloat)aFloat
```

Parameters

aFloat

The amount by which text is inset within line fragment rectangles, in points.

Discussion

This method also sends `textContainerChangedGeometry:` to the text container's layout manager.

Line fragment padding is not designed to express text margins. Instead, use the `NSTextView` method `setTextContainerInset:`, paragraph margin attributes, or the position of the text view within a superview.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- lineFragmentRectForProposedRect:sweepDirection:movementDirection:remainingRect:` (page 10)

`- lineFragmentPadding` (page 10)

Related Sample Code

TipWrapper

Declared In

`NSTextContainer.h`

setTextView:

Sets the receiver's text view.

```
- (void)setTextView:(NSTextView *)aTextView
```

Parameters

aTextView

The new text view.

Discussion

This method sends `setTextContainer:` to *aTextView* to complete the association of the text container and text view.

Because you usually specify a text container when you create a text view, you should rarely need to invoke this method. A text container doesn't need a text view to calculate line fragment rectangles, but must have one to display text.

You can use this method to disconnect a text view from a group of text system objects by sending this message to its text container and passing `nil` as *aTextView*.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- initWithFrame:textContainer:` (NSTextView)

`- replaceTextContainer:` (NSTextView)

Related Sample Code

Sketch-112

Declared In

`NSTextContainer.h`

setWidthTracksTextView:

Controls whether the receiver adjusts the width of its bounding rectangle when its text view is resized.

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

Parameters

flag

YES if the receiver should follow changes to the width of its text view, NO otherwise.

Discussion

See *Text System Storage Layer Overview* for more information on size tracking.

Availability

Available in Mac OS X v10.0 and later.

See Also

- `- setContainerSize:` (page 12)
- `- setHeightTracksTextView:` (page 12)
- `- widthTracksTextView` (page 16)

Related Sample Code

Quartz Composer WWDC 2005 TextEdit

Sketch-112

TextEditPlus

TextLayoutDemo

Declared In

`NSTextContainer.h`

textView

Returns the receiver's text view.

```
- (NSTextView *)textView
```

Return Value

The receiver's text view, or `nil` if it has none.

Availability

Available in Mac OS X v10.0 and later.

See Also

- `- setTextView:` (page 14)

Related Sample Code

Quartz Composer WWDC 2005 TextEdit

TextEditPlus

Declared In
`NSTextContainer.h`

widthTracksTextView

Returns whether the receiver adjusts the width of its bounding rectangle when its text view is resized.

```
- (BOOL)widthTracksTextView
```

Return Value
`YES` if the receiver adjusts the width of its bounding rectangle when its text view is resized, `NO` otherwise.

Discussion
If the receiver does track the text view width, its width is adjusted to the width of the text view minus twice the inset width (as given by `NSTextView`'s `textContainerInset` method).

See *Text System Storage Layer Overview* for more information on size tracking.

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

See Also
`– heightTracksTextView` (page 8)
`– setWidthTracksTextView:` (page 15)

Declared In
`NSTextContainer.h`

Constants

NSLineSweepDirection

These constants describe the progression of text on a page. The typesetter decides which way text is supposed to flow and passes these values as arguments to the text container, which uses them to calculate the next line rectangle.

```
typedef enum {
    NSLineSweepLeft  = 0,
    NSLineSweepRight = 1,
    NSLineSweepDown  = 2,
    NSLineSweepUp    = 3
} NSLineSweepDirection;
```

Constants
`NSLineSweepLeft`

Characters move from right to left.

Available in Mac OS X v10.0 and later.

Declared in `NSTextContainer.h`.

`NSLineSweepRight`

Characters move from left to right.

Available in Mac OS X v10.0 and later.

Declared in `NSTextContainer.h`.

`NSLineSweepDown`

Characters move from top to bottom.

Available in Mac OS X v10.0 and later.

Declared in `NSTextContainer.h`.

`NSLineSweepUp`

Characters move from bottom to top.

Available in Mac OS X v10.0 and later.

Declared in `NSTextContainer.h`.

Discussion

Line sweep is the direction text progresses within a line. See *Text System Storage Layer Overview*.

The only values currently used by the supplied typesetters are `NSLineSweepRight` and `NSLineMovesDown`. An NSTextContainer subclass should be prepared to deal with any value, and an NSTypesetter subclass should be able to use any of them.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSTextContainer.h`

NSLineMovementDirection

Line movement is the direction in which lines move. See *Text System Storage Layer Overview*.

```
typedef enum {
    NSLineDoesntMove = 0,
    NSLineMovesLeft  = 1,
    NSLineMovesRight = 2,
    NSLineMovesDown  = 3,
    NSLineMovesUp    = 4
} NSLineMovementDirection;
```

Constants

`NSLineMovesLeft`

Lines move from right to left.

Available in Mac OS X v10.0 and later.

Declared in `NSTextContainer.h`.

`NSLineMovesRight`

Lines move from left to right.

Available in Mac OS X v10.0 and later.

Declared in `NSTextContainer.h`.

`NSLineMovesDown`

Lines move from top to bottom.

Available in Mac OS X v10.0 and later.

Declared in `NSTextContainer.h`.

`NSLineMovesUp`

Lines move from bottom to top.

Available in Mac OS X v10.0 and later.

Declared in `NSTextContainer.h`.

`NSLineDoesntMove`

Line has no movement.

Available in Mac OS X v10.0 and later.

Declared in `NSTextContainer.h`.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSTextContainer.h`

Document Revision History

This table describes the changes to *NSTextContainer Class Reference*.

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

Index