NSTextTable Class Reference

User Experience: Text Layout

2007-04-11

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

Simultaneously published in the United States and Canada.

Contents

NSTextTable Class Reference

Inherits from	NSTextBlock : NSObject
Conforms to	NSCoding (NSTextBlock) NSCopying (NSTextBlock) NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.4 and later.
Declared in	NSTextTable.h
Companion guides	Text System Overview Text Layout Programming Guide for Cocoa
Related sample code	iSpend

Overview

An `NSTextTable` object represents a text table as a whole. It is responsible for laying out and drawing the text table blocks it contains, and it maintains the basic parameters of the table.

Tasks

Getting and Setting Number of Columns

`- numberOfColumns` (page 9)
Returns the number of columns in the text table.

`- setNumberOfColumns:` (page 11)
Sets the number of columns in the text table.

Getting and Setting Layout Algorithm

`- layoutAlgorithm` (page 8)
Returns the text table layout algorithm.

- `setLayoutAlgorithm:` (page 11)
  Sets the text table layout algorithm.

Collapsing Borders

- `collapsesBorders` (page 7)
  Returns whether the text table borders are collapsible.
- `setCollapsesBorders:` (page 10)
  Sets whether the text table borders are collapsible.

Hiding Empty Cells

- `hidesEmptyCells` (page 8)
  Returns whether the text table hides empty cells.
- `setHidesEmptyCells:` (page 10)
  Sets whether the text table hides empty cells.

Determining Layout Rectangles

- `rectForBlock:layoutAtPoint:inRect:textContainer:characterRange:` (page 9)
  Returns the rectangle within which glyphs should be laid out for a text table block.
- `boundsRectForBlock:contentRect:inRect:textContainer:characterRange:` (page 6)
  Returns the rectangle the text table block actually occupies, including padding, borders, and margins.

Drawing the Table

- `drawBackgroundForBlock:withFrame:inView:characterRange:layoutManager:` (page 7)
  Draws any colors and other decorations for a text table block.

Instance Methods

boundsRectForBlock:contentRect:inRect:textContainer:characterRange:

Returns the rectangle the text table block actually occupies, including padding, borders, and margins.

```
- (NSRect)boundsRectForBlock:(NSTextTableBlock *)block
    contentRect:(NSRect)contentRect inRect:(NSRect)rect
    textContainer:(NSTextContainer *)textContainer characterRange:(NSRange)charRange
```

Parameters

block

The text table block that wants to determine where to layout its glyphs.

contentRect
The actual rectangle in which the text was laid out, as determined by `rectForLayoutAtPoint:inRect:textContainer:characterRange:`.

rect
The initial rectangle in *textContainer* proposed by the typesetter.

textContainer
The text container being used for the layout.

charRange
The range of the characters whose glyphs are to be drawn.

Return Value
The rectangle the text table block actually occupies, including padding, borders, and margins.

Discussion
This method is called by the text table block *block* after it is laid out to determine the rectangle the text table block actually occupies, including padding, borders, and margins.

Availability
Available in Mac OS X v10.4 and later.

See Also
- rectForBlock:layoutAtPoint:inRect:textContainer:characterRange: (page 9)

Declared In
NSTextTable.h

collapsesBorders

Returns whether the text table borders are collapsible.

```
- (BOOL)collapsesBorders
```

Return Value
`YES` if the text table borders are collapsible, `NO` otherwise.

Availability
Available in Mac OS X v10.4 and later.

See Also
- setCollapsesBorders: (page 10)

Declared In
NSTextTable.h

drawBackgroundForBlock:withFrame:inView:characterRange:layoutManager:

Draws any colors and other decorations for a text table block.

```
- (void)drawBackgroundForBlock:(NSTextTableBlock *)block withFrame:(NSRect)frameRect
    inView:(NSView *)controlView characterRange:(NSRange)charRange
   layoutManager:(NSLayoutManager *)layoutManager
```

Parameters

block

The text table block that wants to draw its background.

frameRect

The area in which drawing occurs.

controlView

The view controlling the drawing.

charRange

The range of the characters whose glyphs are to be drawn.

layoutManager

The layout manager controlling the typesetting.

Discussion

This methods is called by the text table block *block* to draw any colors and other decorations before the text is drawn.

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSTextTable.h

hidesEmptyCells

Returns whether the text table hides empty cells.

```
- (BOOL)hidesEmptyCells
```

Return Value

YES if the text table hides empty cells, NO otherwise.

Discussion

If empty cells are hidden, locations with empty cells allow the background of the enclosing block or text container to show through.

Availability

Available in Mac OS X v10.4 and later.

See Also

- setHidesEmptyCells: (page 10)

Declared In

NSTextTable.h

layoutAlgorithm

Returns the text table layout algorithm.

```
- (NSTextTableLayoutAlgorithm)layoutAlgorithm
```

Return Value

The text table layout algorithm.

Availability
Available in Mac OS X v10.4 and later.

See Also
- `setLayoutAlgorithm:` (page 11)

Declared In
`NSTextTable.h`

numberOfColumns

Returns the number of columns in the text table.

```
- (NSUInteger)numberOfColumns
```

Return Value
The number of columns in the text table.

Availability
Available in Mac OS X v10.4 and later.

See Also
- `setNumberOfColumns:` (page 11)

Declared In
`NSTextTable.h`

rectForBlock:layoutAtPoint:inRect:textContainer:characterRange:

Returns the rectangle within which glyphs should be laid out for a text table block.

```
- (NSRect)rectForBlock:(NSTextTableBlock *)block layoutAtPoint:(NSPoint)startingPoint
    inRect:(NSRect)rect textContainer:(NSTextContainer *)textContainer
   characterRange:(NSRange)charRange
```

Parameters

block
The text table block that wants to determine where to layout its glyphs.

startingPoint
The location, in container coordinates, where layout begins.

rect
The rectangle in which the block is constrained to lie. For top-level blocks, this is the container rectangle of *textContainer*; for nested blocks, this is the layout rectangle of the enclosing block.

textContainer
The text container being used for the layout.

charRange
The range of the characters whose glyphs are to be drawn.

Return Value
The rectangle within which glyphs should be laid out.

Discussion
This method is called by the text table block *block* to determine the rectangle within which glyphs should be laid out for the text table block.

Availability
Available in Mac OS X v10.4 and later.

See Also
- boundsRectForBlock:contentRect:inRect:textContainer:characterRange: (page 6)

Declared In
NSTextTable.h

setCollapsesBorders:

Sets whether the text table borders are collapsible.

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

Parameters
flag
YES if the text table borders should be collapsible, NO otherwise.

Availability
Available in Mac OS X v10.4 and later.

See Also
- collapsesBorders (page 7)

Related Sample Code
iSpend

Declared In
NSTextTable.h

setHidesEmptyCells:

Sets whether the text table hides empty cells.

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

Parameters
flag
YES if the text table should hide empty cells, NO otherwise.

Discussion
If empty cells are hidden, locations with empty cells allow the background of the enclosing block or text container to show through.

Availability
Available in Mac OS X v10.4 and later.

See Also
- hidesEmptyCells (page 8)

Related Sample Code
iSpend

Declared In
`NSTextTable.h`

setLayoutAlgorithm:

Sets the text table layout algorithm.

```
- (void)setLayoutAlgorithm:(NSTextTableLayoutAlgorithm)algorithm
```

Parameters

algorithm
The new layout algorithm.

Availability
Available in Mac OS X v10.4 and later.

See Also
- `layoutAlgorithm` (page 8)

Related Sample Code
iSpend

Declared In
`NSTextTable.h`

setNumberOfColumns:

Sets the number of columns in the text table.

```
- (void)setNumberOfColumns:(NSUInteger)numCols
```

Parameters

numCols
The new number of columns.

Availability
Available in Mac OS X v10.4 and later.

See Also
- `numberOfColumns` (page 9)

Related Sample Code
iSpend

Declared In
`NSTextTable.h`

Constants

NSTextTableLayoutAlgorithm

These constants, specifying the type of text table layout algorithm, are used with `setLayoutAlgorithm:` (page 11).

```
enum {
    NSTextTableAutomaticLayoutAlgorithm = 0,
    NSTextTableFixedLayoutAlgorithm     = 1
};
typedef NSUInteger NSTextTableLayoutAlgorithm;
```

Constants

`NSTextTableAutomaticLayoutAlgorithm`

Specifies automatic layout algorithm

Available in Mac OS X v10.4 and later.

Declared in `NSTextTable.h`.

`NSTextTableFixedLayoutAlgorithm`

Specifies fixed layout algorithm

Available in Mac OS X v10.4 and later.

Declared in `NSTextTable.h`.

Declared In

`NSTextTable.h`

Document Revision History

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

Date	Notes
2007-04-11	Reformatted descriptions of constants.
2006-05-23	First publication of this content as a separate document.

Document Revision History

Index