NSSplitView Class Reference

Cocoa > User Experience

2009-04-08

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

NSSplitView Class Reference 5

Overview 5
Tasks 5
Managing Subviews 5
Managing Split View Orientation 6
Assigning a Delegate 6
Configuring and Drawing View Dividers 6
Saving Subview Positions 7
Configuring Pane Splitters 7
Constraining Split Position 7
Instance Methods 7
adjustSubviews 7
autosaveName 8
delegate 8
dividerColor 8
dividerStyle 9
dividerThickness 9
drawDividerInRect: 10
isPaneSplitter 10
isSubviewCollapsed: 10
isVertical 10
maxPossiblePositionOfDividerAtIndex: 11
minPossiblePositionOfDividerAtIndex: 11
setAutosaveName: 12
setDelegate: 12
setDividerStyle: 13
setIsPaneSplitter: 13
setPosition:ofDividerAtIndex: 14
setVertical: 14
Delegate Methods 14
splitView:additionalEffectiveRectOfDividerAtIndex: 14
splitView:canCollapseSubview: 15
splitView:constrainMaxCoordinate:ofSubviewAt: 15
splitView:constrainMinCoordinate:ofSubviewAt: 16
splitView:constrainSplitPosition:ofSubviewAt: 16
splitView:effectiveRect:forDrawnRect:ofDividerAtIndex: 17
splitView:resizeSubviewsWithOldSize: 18
splitView:shouldCollapseSubview:forDoubleClickOnDividerAtIndex: 18
splitView:shouldHideDividerAtIndex: 19
splitViewDidResizeSubviews: 19
splitViewWillResizeSubviews: 20

Constants 20
Split View Divider Styles 20
Notifications 21
NSSplitViewDidResizeSubviewsNotification 21
NSSplitViewWillResizeSubviewsNotification 21

Document Revision History 23

Index 25

NSSplitView Class Reference

Inherits from	NSView : NSResponder : NSObject
Conforms to	NSAnimatablePropertyContainer (NSView) NSCoding (NSResponder) NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.0 and later.
Companion guide	View Programming Guide for Cocoa
Declared in	NSSplitView.h
Related sample code	Link Snoop MyPhoto NewsReader PDFKitLinker2 QTMetadataEditor

Overview

An `NSSplitView` object stacks several subviews within one view so that the user can change their relative sizes. By default, the split bars between the views are horizontal, so the views are one on top of the other.

Divider indices are zero-based, with the topmost (in horizontal split views) or leftmost (vertical) divider having an index of 0.

Tasks

Managing Subviews

`– adjustSubviews` (page 7)
Adjusts the sizes of the receiver's subviews so they (plus the dividers) fill the receiver.

`– isSubviewCollapsed:` (page 10)
Returns `YES` if *`subview`* is in a collapsed state, `NO` otherwise.

`– splitView:resizeSubviewsWithOldSize:` (page 18) *delegate method*
Allows the delegate to specify custom sizing behavior for the subviews of the NSSplitView *`sender`*.

– `splitViewWillResizeSubviews:` (page 20) *delegate method*
Invoked by the default notification center to notify the delegate that the splitview will resize its subviews.

– `splitViewDidResizeSubviews:` (page 19) *delegate method*
Invoked by the default notification center to notify the delegate that the splitview did resize its subviews.

– `splitView:canCollapseSubview:` (page 15) *delegate method*
Allows the delegate to determine whether the user can collapse and uncollapse *`subview`*.

– `splitView:shouldCollapseSubview:forDoubleClickOnDividerAtIndex:` (page 18) *delegate method*
Invoked to allow a delegate to determine if a subview should collapse in response to a double click.

Managing Split View Orientation

– `isVertical` (page 10)
Returns `YES` if the split bars are vertical (subviews are side by side), `NO` if they are horizontal (views are one on top of the other).

– `setVertical:` (page 14)
Sets whether the split bars are vertical.

Assigning a Delegate

– `delegate` (page 8)
Returns the receiver's delegate.

– `setDelegate:` (page 12)
Makes *`anObject`* the receiver's delegate.

Configuring and Drawing View Dividers

– `setDividerStyle:` (page 13)
Sets the style of divider drawn between subviews.

– `dividerStyle` (page 9)
Returns the style of the divider drawn between subviews.

– `dividerThickness` (page 9)
Returns the thickness of the divider.

– `dividerColor` (page 8)
Return the color of the dividers that the split view is drawing between subviews.

– `drawDividerInRect:` (page 10)
Draws the divider between two of the receiver's subviews.

– `splitView:effectiveRect:forDrawnRect:ofDividerAtIndex:` (page 17) *delegate method*
Allows the delegate to modify the rectangle in which mouse clicks initiate divider dragging.

– `splitView:shouldHideDividerAtIndex:` (page 19) *delegate method*
Allows the delegate to determine whether a divider can be dragged or adjusted off the edge of the split view.

- splitView:additionalEffectiveRectOfDividerAtIndex: (page 14) *delegate method*
  Allows the delegate to return an additional rectangle in which mouse clicks will initiate divider dragging.

Saving Subview Positions

- setAutosaveName: (page 12)
  Sets the name under which receiver's divider position is automatically saved.
- autosaveName (page 8)
  Returns the name under which receiver's divider position is automatically saved.

Configuring Pane Splitters

- isPaneSplitter (page 10)
  Returns YES if the receiver's splitter is a bar that goes across the split view. Returns NO if the splitter is a thumb on the regular background pattern.
- setIsPaneSplitter: (page 13)
  Sets the type of splitter.

Constraining Split Position

- minPossiblePositionOfDividerAtIndex: (page 11)
  Returns the minimum possible position of the divider at the specified index.
- maxPossiblePositionOfDividerAtIndex: (page 11)
  Returns the maximum possible position of the divider at the specified index.
- setPosition:ofDividerAtIndex: (page 14)
  Sets the position of the divider at the specified index.
- splitView:constrainMaxCoordinate:ofSubviewAt: (page 15) *delegate method*
  Allows the delegate for *sender* to constrain the maximum coordinate limit of a divider when the user drags it.
- splitView:constrainMinCoordinate:ofSubviewAt: (page 16) *delegate method*
  Allows the delegate for *sender* to constrain the minimum coordinate limit of a divider when the user drags it.
- splitView:constrainSplitPosition:ofSubviewAt: (page 16) *delegate method*
  Allows the delegate for *sender* to constrain the divider to certain positions.

Instance Methods

adjustSubviews

Adjusts the sizes of the receiver's subviews so they (plus the dividers) fill the receiver.

```
- (void)adjustSubviews
```

Discussion

The subviews are resized proportionally; the size of a subview relative to the other subviews doesn't change.

Availability

Available in Mac OS X v10.0 and later.

See Also

- `setDelegate:` (page 12)
- `setFrame:` (NSView)

Related Sample Code

MyPhoto

Declared In

`NSSplitView.h`

autosaveName

Returns the name under which receiver's divider position is automatically saved.

```
- (NSString *)autosaveName
```

Return Value

The name used to save the receiver's state.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `setAutosaveName:` (page 12)

Declared In

`NSSplitView.h`

delegate

Returns the receiver's delegate.

```
- (id)delegate
```

Availability

Available in Mac OS X v10.0 and later.

See Also

- `setDelegate:` (page 12)

Declared In

`NSSplitView.h`

dividerColor

Return the color of the dividers that the split view is drawing between subviews.

```
- (NSColor *)dividerColor
```

Return Value
The color drawn between the subviews.

Discussion
The default implementation of this method returns `clearColor` when `dividerStyle` (page 9) returns `NSSplitViewDividerStyleThick` (page 20). It will also return `clearColor` when `dividerStyle` (page 9) returns `NSSplitViewDividerStyleThin` (page 21) if the split view is in a textured window. All other thin dividers are drawn with a color that looks good between two white panes.

You can override this method to change the color of dividers.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSSplitView.h`

dividerStyle

Returns the style of the divider drawn between subviews.

```
- (NSSplitViewDividerStyle)dividerStyle
```

Return Value
The current divider style. The possible values are described in "Split View Divider Styles" (page 20).

Availability
Available in Mac OS X v10.5 and later.

See Also
`- setDividerStyle:` (page 13)

Declared In
`NSSplitView.h`

dividerThickness

Returns the thickness of the divider.

```
- (CGFloat)dividerThickness
```

Discussion
You can subclass NSSplitView and override this method to change the divider's size, if necessary.

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

See Also
`- drawDividerInRect:` (page 10)

Declared In
`NSSplitView.h`

drawDividerInRect:

Draws the divider between two of the receiver's subviews.

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

Discussion

aRect describes the entire divider rectangle in the receiver's coordinates, which are flipped. If you override this method and use a custom icon to identify the divider, you may need to change the size of the divider.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- dividerThickness` (page 9)

`- compositeToPoint:operation:` (NSImage)

Declared In

`NSSplitView.h`

isPaneSplitter

Returns `YES` if the receiver's splitter is a bar that goes across the split view. Returns `NO` if the splitter is a thumb on the regular background pattern.

```
- (BOOL)isPaneSplitter
```

Availability

Available in Mac OS X v10.0 and later.

See Also

`- setIsPaneSplitter:` (page 13)

Declared In

`NSSplitView.h`

isSubviewCollapsed:

Returns `YES` if *subview* is in a collapsed state, `NO` otherwise.

```
- (BOOL)isSubviewCollapsed:(NSView *)subview
```

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSSplitView.h`

isVertical

Returns `YES` if the split bars are vertical (subviews are side by side), `NO` if they are horizontal (views are one on top of the other).

```
- (BOOL)isVertical
```

Discussion

By default, split bars are vertical.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- setVertical:` (page 14)

Declared In

`NSSplitView.h`

maxPossiblePositionOfDividerAtIndex:

Returns the maximum possible position of the divider at the specified index.

```
- (CGFloat)maxPossiblePositionOfDividerAtIndex:(NSInteger)dividerIndex
```

Parameters

dividerIndex

The index of the divider.

Return Value

A CGFloat specifying the maximum possible position of the divider.

Discussion

The position is "possible" in that it is dictated by the bounds of the receiver and the current position of other dividers. "Allowable" positions are those that result from letting the delegate apply constraints to the possible positions.

You can invoke this method to determine the range of values that can be usefully passed to `setPosition:ofDividerAtIndex:` (page 14).

You can also invoke it from delegate methods like `splitView:constrainSplitPosition:ofSubviewAt:` (page 16) to implement relatively complex behaviors that depend on the current state of the split view.

The results of invoking this method when `adjustSubviews` (page 7) has not been invoked, and the subview frames are invalid, is undefined.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`NSSplitView.h`

minPossiblePositionOfDividerAtIndex:

Returns the minimum possible position of the divider at the specified index.

```
- (CGFloat)minPossiblePositionOfDividerAtIndex:(NSInteger)dividerIndex
```

Parameters

dividerIndex

The index of the divider.

Return Value

A CGFloat specifying the minimum possible position of the divider.

Discussion

The position is "possible" in that it is dictated by the bounds of the receiver and the current position of other dividers. "Allowable" positions are those that result from letting the delegate apply constraints to the possible positions.

You can invoke this method to determine the range of values that can be usefully passed to `setPosition:ofDividerAtIndex:` (page 14).

You can also invoke it from delegate methods like `splitView:constrainSplitPosition:ofSubviewAt:` (page 16) to implement relatively complex behaviors that depend on the current state of the split view.

The results of invoking this method when `adjustSubviews` (page 7) has not been invoked, and the subview frames are invalid, is undefined.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`NSSplitView.h`

setAutosaveName:

Sets the name under which receiver's divider position is automatically saved.

```
- (void)setAutosaveName:(NSString *)autosaveName
```

Parameters

autosaveName

The name used to save the receiver's state.

Discussion

If this value is `nil` or the string is empty no autosaving is done.

Availability

Available in Mac OS X v10.5 and later.

See Also

`– autosaveName` (page 8)

Declared In

`NSSplitView.h`

setDelegate:

Makes *anObject* the receiver's delegate.

```
- (void)setDelegate:(id)anObject
```

Discussion

The notification messages the delegate can expect to receive are listed in "Notifications" (page 21). The delegate doesn't need to implement all of the delegate methods.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- delegate` (page 8)

Declared In

`NSSplitView.h`

setDividerStyle:

Sets the style of divider drawn between subviews.

```
- (void)setDividerStyle:(NSSplitViewDividerStyle)dividerStyle
```

Parameters

`dividerStyle`

The divider style. Possible values are described in "Split View Divider Styles" (page 20).

Availability

Available in Mac OS X v10.5 and later.

See Also

`- dividerStyle` (page 9)

Declared In

`NSSplitView.h`

setIsPaneSplitter:

Sets the type of splitter.

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

Discussion

If *`flag`* is `YES`, the receiver's splitter is a bar that goes across the split view. If *`flag`* is `NO`, the splitter is a thumb on the regular background pattern.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- isPaneSplitter` (page 10)

Declared In

`NSSplitView.h`

setPosition:ofDividerAtIndex:

Sets the position of the divider at the specified index.

```
- (void)setPosition:(CGFloat)position
    ofDividerAtIndex:(NSInteger)dividerIndex
```

Parameters

position

The position of the divider

dividerIndex

The index of the divider.

Discussion

The default implementation of this method behaves as if the user were attempting to drag the divider to the proposed position, so the constraints imposed by the delegate are applied and one of the views adjacent to the divider may be collapsed.

This method is not invoked by `NSSplitView` itself.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`NSSplitView.h`

setVertical:

Sets whether the split bars are vertical.

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

Discussion

If *flag* is `YES`, they're vertical (views are side by side); if it's `NO`, they're horizontal (views are one on top of the other). Split bars are horizontal by default.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- isVertical` (page 10)

Declared In

`NSSplitView.h`

Delegate Methods

splitView:additionalEffectiveRectOfDividerAtIndex:

Allows the delegate to return an additional rectangle in which mouse clicks will initiate divider dragging.

```
- (NSRect)splitView:(NSSplitView *)splitView
    additionalEffectiveRectOfDividerAtIndex:(NSInteger)dividerIndex
```

Parameters

splitView

The split view that sent the message.

dividerIndex

The index of the divider.

Return Value

An additional rectangle in which mouse clicks should initiate divider dragging. The rectangle should be expressed in the coordinate system defined by *splitView*. Returning NSZeroRect indicates no additional dragging rectangle is desired.

Discussion

If a split view has no delegate, or if its delegate does not respond to this message, only mouse clicks within the effective frame of a divider initiate divider dragging.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSSplitView.h

splitView:canCollapseSubview:

Allows the delegate to determine whether the user can collapse and uncollapse *subview*.

```
- (BOOL)splitView:(NSSplitView *)sender canCollapseSubview:(NSView *)subview
```

Discussion

If this method returns NO or is not implemented, *subview* can't be collapsed. If this method returns YES, *subview* collapses when the user drags a divider beyond the halfway mark between its minimum size and its edge. *subview* uncollapses when the user drags the divider back beyond that point. To specify the minimum size, define the methods splitView:constrainMaxCoordinate:ofSubviewAt: (page 15) and splitView:constrainMinCoordinate:ofSubviewAt: (page 16). Note that a subview can collapse only if you also define splitView:constrainMinCoordinate:ofSubviewAt: (page 16).

A collapsed subview is hidden but retained by the NSSplitView object, with the same size it had before it was collapsed.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSSplitView.h

splitView:constrainMaxCoordinate:ofSubviewAt:

Allows the delegate for *sender* to constrain the maximum coordinate limit of a divider when the user drags it.

```
- (CGFloat)splitView:(NSSplitView *)sender
    constrainMaxCoordinate:(CGFloat)proposedMax
    ofSubviewAt:(NSInteger)offset
```

Discussion
This method is invoked before the NSSplitView begins tracking the mouse to position a divider. You may further constrain the limits that have been already set, but you cannot extend the divider limits. *proposedMax* is specified in the NSSplitView's flipped coordinate system. If the split bars are horizontal (views are one on top of the other), *proposedMax* is the bottom limit. If the split bars are vertical (views are side by side), *proposedMax* is the right limit. The initial value of *proposedMax* is the bottom (or right side) of the subview after the divider. *offset* specifies the divider the user is moving, with the first divider being 0 and going up from top to bottom (or left to right).

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

See Also
`- isVertical` (page 10)

Declared In
`NSSplitView.h`

splitView:constrainMinCoordinate:ofSubviewAt:

Allows the delegate for *sender* to constrain the minimum coordinate limit of a divider when the user drags it.

```
- (CGFloat)splitView:(NSSplitView *)sender
    constrainMinCoordinate:(CGFloat)proposedMin
    ofSubviewAt:(NSInteger)offset
```

Discussion
This method is invoked before the NSSplitView begins tracking the cursor to position a divider. You may further constrain the limits that have been already set, but you cannot extend the divider limits. *proposedMin* is specified in the NSSplitView's flipped coordinate system. If the split bars are horizontal (views are one on top of the other), *proposedMin* is the top limit. If the split bars are vertical (views are side by side), *proposedMin* is the left limit. The initial value of *proposedMin* is the top (or left side) of the subview before the divider. *offset* specifies the divider the user is moving, with the first divider being 0 and going up from top to bottom (or left to right).

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

See Also
`- isVertical` (page 10)

Declared In
`NSSplitView.h`

splitView:constrainSplitPosition:ofSubviewAt:

Allows the delegate for *sender* to constrain the divider to certain positions.

```
- (CGFloat)splitView:(NSSplitView *)sender
    constrainSplitPosition:(CGFloat)proposedPosition
    ofSubviewAt:(NSInteger)offset
```

Discussion
If the delegate implements this method, the NSSplitView calls it repeatedly as the user moves the divider. This method returns where you want the divider to be, given *proposedPosition*, the cursor's current position. *offset* is the divider the user is moving, with the first divider being 0 and going up from top to bottom (or from left to right).

For example, if a subview's height must be a multiple of a certain number, use this method to return the multiple nearest to *proposedPosition*.

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

Declared In
`NSSplitView.h`

splitView:effectiveRect:forDrawnRect:ofDividerAtIndex:

Allows the delegate to modify the rectangle in which mouse clicks initiate divider dragging.

```
- (NSRect)splitView:(NSSplitView *)splitView
    effectiveRect:(NSRect)proposedEffectiveRect
    forDrawnRect:(NSRect)drawnRect
    ofDividerAtIndex:(NSInteger)dividerIndex
```

Parameters

splitView
  The split view that sent the message.

proposedEffectiveRect
  The proposed rectangle in which mouse clicks should initiate divider dragging. The rectangle is expressed in the coordinate system defined by *splitView*.

drawnRect
  The frame of the divider, expressed in the coordinate system defined by *splitView*.

dividerIndex
  The index of the divider.

Return Value
The rectangle in which mouse clicks should initiate divider dragging. The rectangle should be expressed in the coordinate system defined by *splitView*.

Discussion
A split view with thick dividers proposes the drawn frame as the effective frame. A split view with thin dividers proposes an effective frame that's a little larger than the drawn frame, to make it easier for the user to actually grab the divider.

If a split view has no delegate, or if its delegate does not respond to this message, the split view behaves as if it has a delegate that returns *proposedEffectiveRect* when sent this message.

Availability
Available in Mac OS X v10.5 and later.

Declared In

`NSSplitView.h`

splitView:resizeSubviewsWithOldSize:

Allows the delegate to specify custom sizing behavior for the subviews of the NSSplitView *sender*.

```
- (void)splitView:(NSSplitView *)sender
    resizeSubviewsWithOldSize:(NSSize)oldSize
```

Discussion

If the delegate implements this method, `splitView:resizeSubviewsWithOldSize:` is invoked after the NSSplitView is resized. The size of the NSSplitView before the user resized it is indicated by *oldSize*; the subviews should be resized such that the sum of the sizes of the subviews plus the sum of the thickness of the dividers equals the size of the NSSplitView's new frame. You can get the thickness of a divider through the *dividerThickness* method.

Note that if you implement this delegate method to resize subviews on your own, the NSSplitView does not perform any error checking for you. However, you can invoke `adjustSubviews` (page 7) to perform the default sizing behavior.

Availability

Available in Mac OS X v10.0 and later.

See Also

- `adjustSubviews` (page 7)
- `setFrame:` (NSView)

Declared In

`NSSplitView.h`

splitView:shouldCollapseSubview:forDoubleClickOnDividerAtIndex:

Invoked to allow a delegate to determine if a subview should collapse in response to a double click.

```
- (BOOL)splitView:(NSSplitView *)splitView shouldCollapseSubview:(NSView *)subview
      forDoubleClickOnDividerAtIndex:(NSInteger)dividerIndex
```

Parameters

splitView

The splitview that sent the message.

subview

The subview to collapse.

dividerIndex

The index of the divider.

Return Value

`YES` if the subview should collapse, `NO` otherwise.

Discussion
If implemented, the delegate will receive this message once for the subview before a divider when the user double-clicks on that divider, and again for the subview after the divider, but only if the delegate returned `YES` when sent `splitView:canCollapseSubview:` (page 15) for the subview in question. When the delegate indicates that both subviews should be collapsed NSSplitView's behavior is undefined.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSSplitView.h`

splitView:shouldHideDividerAtIndex:

Allows the delegate to determine whether a divider can be dragged or adjusted off the edge of the split view.

```
- (BOOL)splitView:(NSSplitView *)splitView
    shouldHideDividerAtIndex:(NSInteger)dividerIndex
```

Parameters

splitView
The split view that sent the message.

dividerIndex
The zero-based index of the divider.

Return Value
`YES` if the divider should allow dragging off the edge of the split view, resulting in it not being visible.

Discussion
If a split view has no delegate, or if its delegate does not respond to this message, the split view behaves as if it has a delegate that returns `NO` when sent this message.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSSplitView.h`

splitViewDidResizeSubviews:

Invoked by the default notification center to notify the delegate that the splitview did resize its subviews.

```
- (void)splitViewDidResizeSubviews:(NSNotification *)aNotification
```

Parameters

aNotification
An `NSSplitViewDidResizeSubviewsNotification` (page 21) notification.

Discussion
If the delegate implements this method, the delegate is automatically registered to receive this notification.

This method is invoked after the NSSplitView resizes two of its subviews in response to the repositioning of a divider.

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

Declared In
`NSSplitView.h`

splitViewWillResizeSubviews:

Invoked by the default notification center to notify the delegate that the splitview will resize its subviews.

```
- (void)splitViewWillResizeSubviews:(NSNotification *)aNotification
```

Parameters

aNotification
An `NSSplitViewWillResizeSubviewsNotification` (page 21) notification.

Discussion
If the delegate implements this method, the delegate is automatically registered to receive this notification.

This method is invoked before the NSSplitView resizes two of its subviews in response to the repositioning of a divider.

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

Declared In
`NSSplitView.h`

Constants

Split View Divider Styles

These constants specify the possible divider styles used by `dividerStyle` (page 9) and `setDividerStyle:` (page 13).

```
enum {
    NSSplitViewDividerStyleThick = 1,
    NSSplitViewDividerStyleThin = 2,
};
typedef NSInteger NSSplitViewDividerStyle;
```

Constants

`NSSplitViewDividerStyleThick`
A thick style divider is displayed between subviews. This is the default.

Available in Mac OS X v10.5 and later.

Declared in `NSSplitView.h`.

`NSSplitViewDividerStyleThin`
A thin style divider is displayed between subviews.

Available in Mac OS X v10.5 and later.

Declared in `NSSplitView.h`.

Notifications

NSSplitView declares and posts the following notifications. In addition, it posts notifications declared by its superclass, NSView. See the `NSView` class specification for more information.

NSSplitViewDidResizeSubviewsNotification

Posted after an NSSplitView changes the sizes of some or all of its subviews. The notification object is the NSSplitView that resized its subviews.

Note: In Mac OS X v10.5 and later if the notification is sent because the user is dragging a divider, the `userInfo` dictionary contains a key `@"NSSplitViewDividerIndex"` containing an `NSInteger`-wrapped `NSNumber` that is the index of the divider being dragged. Earlier versions of Mac OS X do not return a user info dictionary in any situation.

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

See Also
`– splitViewDidResizeSubviews:` (page 19)

Declared In
`NSSplitView.h`

NSSplitViewWillResizeSubviewsNotification

Posted before an NSSplitView changes the sizes of some or all of its subviews. The notification object is the NSSplitView object that is about to resize its subviews.

Note: In Mac OS X v10.5 and later if the notification is sent because the user is dragging a divider, the `userInfo` dictionary contains a key `@"NSSplitViewDividerIndex"` containing an `NSInteger`-wrapped `NSNumber` that is the index of the divider being dragged. Earlier versions of Mac OS X do not return a user info dictionary in any situation.

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

See Also
`– splitViewWillResizeSubviews:` (page 20)

Declared In
`NSSplitView.h`

Document Revision History

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

Date	Notes
2009-04-08	Added splitView:shouldHideDividerAtIndex: delegate method. Added divider style and color methods.
2007-06-13	Updated for Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.

Index

A

adjustSubviews instance method 7
autosaveName instance method 8

D

delegate instance method 8
dividerColor instance method 8
dividerStyle instance method 9
dividerThickness instance method 9
drawDividerInRect: instance method 10

I

isPaneSplitter instance method 10
isSubviewCollapsed: instance method 10
isVertical instance method 10

M

maxPossiblePositionOfDividerAtIndex: instance method 11
minPossiblePositionOfDividerAtIndex: instance method 11

N

NSSplitViewDidResizeSubviewsNotification notification 21
NSSplitViewDividerStyleThick constant 20
NSSplitViewDividerStyleThin constant 21
NSSplitViewWillResizeSubviewsNotification notification 21

S

setAutosaveName: instance method 12
setDelegate: instance method 12
setDividerStyle: instance method 13
setIsPaneSplitter: instance method 13
setPosition:ofDividerAtIndex: instance method 14
setVertical: instance method 14
Split View Divider Styles 20
splitView:additionalEffectiveRectOfDividerAtIndex: <NSObject> delegate method 14
splitView:canCollapseSubview: <NSObject> delegate method 15
splitView:constrainMaxCoordinate:ofSubviewAt: <NSObject> delegate method 15
splitView:constrainMinCoordinate:ofSubviewAt: <NSObject> delegate method 16
splitView:constrainSplitPosition:ofSubviewAt: <NSObject> delegate method 16
splitView:effectiveRect:forDrawnRect:ofDividerAtIndex: <NSObject> delegate method 17
splitView:resizeSubviewsWithOldSize: <NSObject> delegate method 18
splitView:shouldCollapseSubview:forDoubleClickOnDividerAtIndex: <NSObject> delegate method 18
splitView:shouldHideDividerAtIndex: <NSObject> delegate method 19
splitViewDidResizeSubviews: <NSObject> delegate method 19
splitViewWillResizeSubviews: <NSObject> delegate method 20