NSBox

Inherits From:	NSView : NSResponder : NSObject
Conforms To:	NSCoding (from NSResponder) NSObject (from NSObject)
Declared In:	AppKit/NSBox.h

Class Description

An NSBox object is a simple NSView that can do two things: It can draw a border around itself and it can title itself. You can use an NSBox to group, visually, some number of other NSViews. These other NSViews are added to the NSBox through the typical subview-adding methods, such as **addSubview:** and **replaceSubview:with:**.

An NSBox contains a *content area*, a rectangle set within the NSBox's frame in which the NSBox's subviews are displayed. The size and location of the content area depends on the NSBox's border type, title location, the size of the font used to draw the title, and an additional measure that you can set through the **setContentViewMargins:** method. When you create an NSBox, an instance of NSView is created and added (as a subview of the NSBox object) to fill the NSBox's content area. If you replace this *content view* with an NSView of your own, your NSView will be resized to fit the content area. Similarly, as you resize an NSBox its content view is automatically resized to fill the content area.

The NSViews that you add as subviews to an NSBox are actually added to the NSBox's content view—NSView's subview-adding methods are redefined by NSBox to ensure that a subview is correctly placed in the view hierarchy. However, you should note that the **subviews** method *isn't* redefined: It returns an NSArray containing a single object, the NSBox's content view.

Method Types

Getting and modifying the border and title	
	– borderRect
	– borderType
	– setBorderType:
	– setTitle:
	– setTitleFont:
	– setTitlePosition:
	– setTitleWithMnemonic:
	– title
	– titleCell
	– titleFont
	– titlePosition
	– titleRect
Setting and placing the content view	
	– contentView
	– contentViewMargins
	– setContentView:
	– setContentViewMargins:
Resizing the box	– setFrameFromContentFrame:
	– sizeToFit

Instance Methods

borderRect

– (NSRect)**borderRect**

Returns the rectangle in which the border is drawn.

borderType

– (NSBorderType)**borderType**

Returns the NSBox's border type. Border types are defined in **NSView.h**; currently, the following border types are defined:

NSNoBorder
NSLineBorder
NSBezelBorder
NSGrooveBorder

By default, an NSBox's border type is NSGrooveBorder.

contentView

– (id)**contentView**

Returns the NSBox's content view. The content view is created automatically when the box is created, and resized as the box is resized (you should never send frame-altering messages directly to a box's content view). You can replace it with an NSView of your own through the **setContentView:** method.

contentViewMargins

– (NSSize)**contentViewMargins**

Returns the distances between the border and the content view. By default, on Mach systems both the width (the horizontal distance between the innermost edge of the border and the content view) and the height (the vertical distance between the innermost edge of the border and the content view) of the returned NSSize are 5.0 in the box's coordinate system.

setBorderType:

– (void)**setBorderType:**(NSBorderType)*aType*

Sets the border type to *aType*, which must be a valid border type. Border types are defined in **NSView.h**; currently, the following border types are defined:

NSNoBorder
NSLineBorder
NSBezelBorder
NSGrooveBorder

If the size of the new border is different from that of the old border, the content view is resized to absorb the difference and the box is marked for redisplay.

See also: **– setNeedsDisplay:** (NSView)

setContentView:

– (void)**setContentView:**(NSView *)*aView*

Sets the NSBox's content view to *aView*, resizing the NSView to fit within the box's current content area. The box is marked for redisplay.

See also: **– setFrameFromContentFrame:**, **– sizeToFit:**, **– setNeedsDisplay:** (NSView)

setContentViewMargins:

– (void)**setContentViewMargins:**(NSSize)*offsetSize*

Sets the horizontal and vertical distance between the border of the NSBox and its content view. The *horizontal* value is applied (reckoned in the box's coordinate system) fully and equally to the left and right sides of the box. The *vertical* value is similarly applied to the top and bottom.

Unlike changing a box's other attributes, such as its title position or border type, changing the offsets *doesn't* automatically resize the content view. In general, you should send a **sizeToFit** message to the box after changing the size of its offsets. This causes the content view to remain unchanged while the box is sized to fit around it.

setFrameFromContentFrame:

– (void)**setFrameFromContentFrame:**(NSRect)*contentFrame*

Places the NSBox so its content view lies on *contentFrame*, reckoned in the coordinate system of the box's superview. The box is marked for redisplay.

See also: – **setContentViewMargins:**, – **setFrame:** (NSView), **– setNeedsDisplay** (NSView)

setTitle:

– (void)**setTitle:**(NSString *)*aString*

Sets the title to *aString*, and marks the region of the receiver within the title rectangle as needing display. By default, an NSBox's title is "Title". If the size of the new title is different from that of the old title, the content view is resized to absorb the difference.

See also: **– setNeedsDisplayInRect:** (NSView), **– titleRect**

setTitleFont:

– (void)**setTitleFont:**(NSFont *)*aFont*

Sets *aFont* as the NSFont object used to draw the NSBox's title, and marks the region of the receiver within the title rectangle as needing display. On Mach systems the title is drawn using the 12.0 point system font by default. If the size of the new font is different from that of the old font, the content view is resized to absorb the difference.

See also: **– setNeedsDisplayInRect:** (NSView)

setTitlePosition:

– (void)**setTitlePosition:**(NSTitlePosition)*aPosition*

Sets the title position to *aPosition*, which can be one of the values listed in the following table. The default position is NSAtTop.

Value	**Meaning**
NSNoTitle	The box has no title
NSAboveTop	Title positioned above the box’s top border
NSAtTop	Title positioned within the box’s top border
NSBelowTop	Title positioned below the box’s top border
NSAboveBottom	Title positioned above the box’s bottom border
NSAtBottom	Title positioned within the box’s bottom border
NSBelowBottom	Title positioned below the box’s bottom border

If the new title position changes the size of the box’s border area, the content view is resized to absorb the difference, and the box is marked as needing redisplay.

See also: **– setNeedsDisplay:** (NSView)

setTitleWithMnemonic:

– (void)**setTitleWithMnemonic:**(NSString *)*aString*

Sets the title to *aString*, taking into account the fact that an embedded “&” character is not a literal but instead marks the title’s “mnemonic.” The character immediately following the “&” character will be underlined.

By default, an NSBox’s title is “Title”. The content view is not automatically resized, and the box is not marked for redisplay.

See also: **– setTitleWithMnemonic:** (NSCell)

sizeToFit

– (void)**sizeToFit**

Resizes and moves the NSBox’s content view so that it just encloses its subviews. The box itself is then moved and resized to wrap around the content view. The box’s width is constrained so its title will be fully displayed.

You should invoke this method after:

- Adding a subview (to the content view).
- Altering the size or location of such a subview.
- Setting the margins around the content view.

The mechanism by which the content view is moved and resized depends on whether the object responds to its own **sizeToFit** message: If it does respond, then that message is sent and the content view is expected to be so modified. If the content view doesn't respond, the box moves and resizes the content view itself.

title

– (NSString *)**title**

Returns the NSBox's title. By default, a box's title is "Title".

titleCell

– (id)**titleCell**

Returns the NSCell that's used to display the NSBox's title.

titleFont

– (NSFont *)**titleFont**

Returns the NSFont that's used to draw the NSBox's title. On Mach systems the title is drawn using the 12.0 point system font by default.

titlePosition

– (NSTitlePosition)**titlePosition**

Returns a constant representing the title position. See the description of **setTitlePosition:** for a list of the title position constants.

titleRect

– (NSRect)**titleRect**

Returns the rectangle in which the NSBox's title is drawn.

See also: **– setTitlePosition**, **– setTitle**, **– setTitleFont**, **– setFrameFromContentFrame:**, **– sizeToFit**