Boxes

User Experience: Windows & Views

2003-02-15

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

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

Figures and Listings

Setting a Box's Border Appearance 11

Using a Box to Create a Separator 15

Introduction to Boxes

A box is a simple view that can draw a border around itself and title itself. It visually groups together other views, including controls.

Organization of This Document

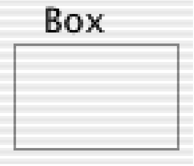
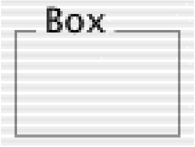
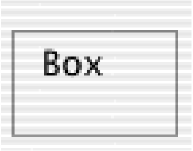
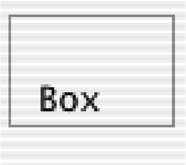
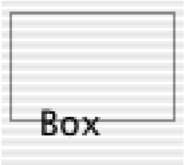
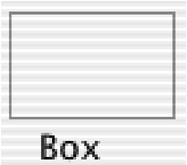
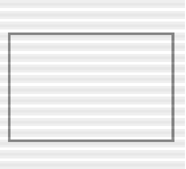
This topic discusses how to use the NSBox class.

- "Setting a Box's Subviews" (page 13) describes how to add subviews to a box.
- "Setting a Box's Title" (page 9) describes how to change the content and appearance of the box's title.
- "Setting a Box's Border Appearance" (page 11) describes how to change the appearance of the box's border.
- "Using a Box to Create a Separator" (page 15) describes how to create visual separators similar to Carbon's theme separator.

If you want to group several elements of the same class together, see Matrices.

Setting a Box’s Title

You can have a box with no title or with a title positioned in one of six places. To set the title’s location, use `setTitlePosition:` with one of options listed below. By default, it’s `NSAtTop`.

`NSAboveTop` 	`NSAtTop` 	`NSBelowTop` 
`NSAboveBottom` 	`NSAtBottom` 	`NSBelowBottom` 
`NSNoTitle` 		

To set the title’s font, use `setTitleFont:`. By default, it’s the Control Content font, which you can set in the Fonts tab of the Appearance panel in the Preferences application. Note that the title appears inside the box, so changing the size of the font will affect the size of the box’s content rectangle.

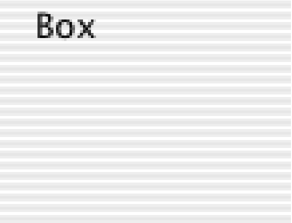
To set the title, use `setTitle:`. By default, it’s “Title”. If the title is longer than the box’s width, the title is clipped.

Setting a Box's Border Appearance

The appearance of an NSBox is set using a combination of the methods `setBoxType:` and `setBorderType:`. By default an NSBox instance is set to use the `NSBoxPrimary` box type and the `NSGrooveBorder` border type.

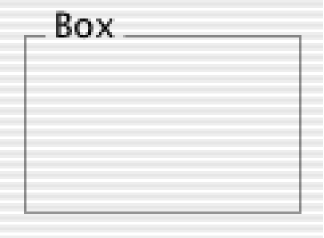
The appearance in Figure 1 is achieved by calling the method `setBorderType:` and specifying `NSNoBorder` as the box type.

Figure 1 NSBox with no visible border



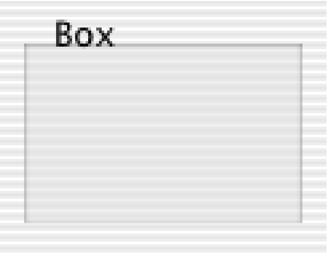
The appearance in Figure 2 is achieved by calling the method `setBorderType:` and specifying `NSGrooveBorder` as the border type, and calling `setBoxType:` and specifying `NSBoxPrimary` as the box type.

Figure 2 NSBox displaying primary appearance



The appearance in Figure 3 is achieved by calling the method `setBorderType:` and specifying `NSGrooveBorder` as the border type, and calling `setBoxType:` and specifying `NSBoxSecondary` as the box type.

Figure 3 NSBox displaying secondary appearance



Note that the border appears inside the box and can reduce the amount of space available to the content rectangle.

Setting a Box's Subviews

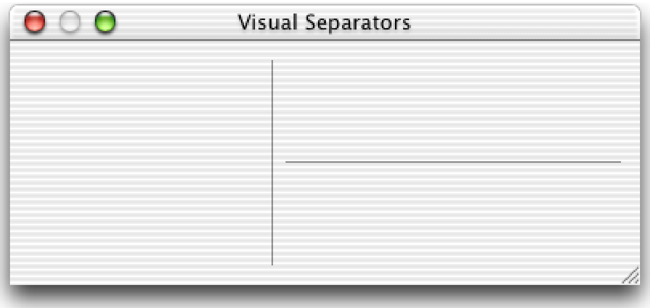
To add a subview to a box, use the typical subview-adding methods, such as `addSubview:` and `replaceSubview:with:`. These NSViews are actually added to the NSBox's content area, an NSView set within the NSBox's frame. NSView's subview-adding methods are redefined by NSBox to ensure that a subview is 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.

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.

Using a Box to Create a Separator

An NSBox instance can be used to create a visual separator between controls, as shown in Figure 1. This is the Cocoa equivalent of the Carbon theme separator.

Figure 1 Examples of vertical and horizontal separators



Calling the method `setBoxType:` and specifying `NSBoxSeparator` as the box type will configure the receiving `NSBox` instance to display as a separator.

The separator will be drawn centered in the view, and oriented along the longest axis of the view. If the bounds of the `NSBox` are equal, then the separator is drawn in a horizontal orientation.

The example code in Listing 1 demonstrates how to create visual separators.

Listing 1 Example code to create visual separators

```
// create a horizontally oriented separator
NSBox *horizontalSeparator=[[NSBox alloc]
initWithFrame:NSMakeRect(15.0,250.0,250.0,1.0)];
[horizontalSeparator setBoxType:NSBoxSeparator];

// create a vertically oriented separator
NSBox *verticalSeparator=[[NSBox alloc]
initWithFrame:NSMakeRect(250.0,15.0,1.0,250.0)];
[verticalSeparator setBoxType:NSBoxSeparator];
```

Document Revision History

This table describes the changes to *Boxes*.

Date	Notes
2003-02-15	Updated "Setting a Box's Border Appearance" (page 11) to reflect Aqua Human Interface Guidelines.
2003-02-12	Added "Using a Box to Create a Separator" (page 15) task.
2002-11-12	Revision history was added to existing topic. It will be used to record changes to the content of the topic.