UISplitViewController Class Reference

User Experience: Windows & Views

2010-05-20

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

UISplitViewController Class Reference

Inherits from	UIViewController : UIResponder : NSObject
Conforms to	NSCoding (UIViewController) NSObject (NSObject)
Framework	/System/Library/Frameworks/UIKit.framework
Availability	Available in iOS 3.2 and later.
Companion guide	iPad Programming Guide
Declared in	UISplitViewController.h
Related sample code	MultipleDetailViews

Overview

The `UISplitViewController` class is a container view controller that manages the presentation of two side-by-side view controllers. You use this class to implement a master-detail interface, in which the left-side view controller presents a list of items and the right-side presents details of the selected item. Split view controllers are for use exclusively on iPad devices. Attempting to create one on other devices results in an exception.

After creating and initializing an instance of this class, you must assign two view controllers to the `viewControllers` (page 7) property. The split view controller has no significant interface of its own. Its job is to coordinate the presentation of its two child view controllers and to manage the transitions among different orientations.

A split view controller supports the same interface orientations as its currently visible child view controllers. Both view controllers are displayed in landscape orientations but only the detail view controller is displayed in portrait orientations. When transitioning between orientations, the split view controller sends messages to its `delegate` (page 6) object to coordinate the display of a popover with the hidden view controller. For more information on the methods of this delegate object, see *UISplitViewControllerDelegate Protocol Reference*.

Note: A split view controller does not provide any inherent support for managing the communication between the custom view controllers you assign to it. It is your responsibility to determine the best way to do that. However, the delegation pattern often works well for master-detail interfaces. To implement such a pattern, your master view controller sends messages to a custom delegate object whenever the selected item changed or some other relevant event occurred. The detail view controller would then assign itself as the delegate of the master and would use the associated messages to refresh its contents.

Message Forwarding to Its Child View Controllers

A split view controller interposes itself between the application's window and its child view controllers. As a result, all messages to the visible view controllers must flow through the split view controller. This works generally as you might expect and the flow of messages should be relatively intuitive. For example, view appearance and disappearance messages are sent only when the corresponding child view controller actually appears on screen. Thus, when a split view controller is first displayed in a portrait orientation, it calls the `viewWillAppear:` and `viewDidAppear:` methods of only the view controller that is shown initially. The view controller that is presented using a popover does not receive those messages until the popover is shown or until the split view controller rotates to a landscape orientation.

Tasks

Managing the Child View Controllers

`viewControllers` (page 7) *property*
The array of view controllers managed by the receiver.

Accessing the Delegate Object

`delegate` (page 6) *property*
The delegate you want to receive split view controller messages.

Properties

For more about Objective-C properties, see "Properties" in *The Objective-C Programming Language*.

delegate

The delegate you want to receive split view controller messages.

```
@property(nonatomic, assign) id <UISplitViewControllerDelegate> delegate
```

Discussion
The split view controller uses its delegate to manage the showing and hiding of related view controllers. For more information about the methods you can implement in your delegate, see *UISplitViewControllerDelegate Protocol Reference*.

Availability
Available in iOS 3.2 and later.

Declared In
`UISplitViewController.h`

viewControllers

The array of view controllers managed by the receiver.

```
@property(nonatomic, copy) NSArray *viewControllers
```

Discussion
The array in this property must contain exactly two view controllers. The view controllers are presented left-to-right in the split view interface when it is in a landscape orientation. Thus, the view controller at index 0 is displayed on the left side and the view controller at index 1 is displayed on the right side of the interface.

The first view controller in this array is typically hidden when the device is in a portrait orientation. Assign a delegate object to the receiver if you want to coordinate the display of this view controller using a popover.

Availability
Available in iOS 3.2 and later.

Related Sample Code
MultipleDetailViews

Declared In
`UISplitViewController.h`

Document Revision History

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

Date	Notes
2010-05-20	Fixed a fragment sentence in the delegate property discussion.
2010-03-12	New document describing a view controller that presents content in left and right panes.