# UISearchDisplayController Class Reference

**User Experience**

**2009-08-13**

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

# UISearchDisplayController Class Reference

| | |
|---|---|
| **Inherits from** | NSObject |
| **Conforms to** | NSObject (NSObject) |
| **Framework** | /System/Library/Frameworks/UIKit.framework |
| **Availability** | Available in iOS 3.0 and later. |
| **Declared in** | UISearchDisplayController.h |

## Overview

A search display controller manages display of a search bar and a table view that displays the results of a search of data managed by another view controller.

You initialize a search display controller with a search bar and a view controller responsible for managing the original content to be searched. When the user starts a search, the search display controller is responsible for superimposing the search interface over the original view controller's view and showing the search results. The results are displayed in a table view that's created by the search display controller. In addition to the original view controller, there are logically four other roles. These are typically all be played by the same object, often the original view controller itself.

1. The search results table view's data source.

   This object is responsible for providing the data for the results table.

2. The search results table view's delegate.

   This object is responsible for, amongst other things, responding to the user's selection of an item in the results table.

3. The search display controller's delegate.

   The delegate conforms to the `UISearchDisplayDelegate` protocol. It is notified of events such as when the search starts or ends, and when the search interface is displayed or hidden. As a convenience, it may also be told about changes to the search string or search scope, so that the results table view can be reloaded.

4. The search bar's delegate.

   This object is responsible for responding to changes in the search criteria.

Typically you initialize a search display controller from a view controller (usually an instance of `UITableViewController`) that's displaying a list; you set `self` for the search display controller's view controller and search results data source and delegate:

```
searchController = [[UISearchDisplayController alloc]
                         initWithSearchBar:searchBar contentsController:self];
searchController.delegate = self;
searchController.searchResultsDataSource = self;
searchController.searchResultsDelegate = self;
```

If you follow this pattern, then in the table view data source and delegate methods you can check the methods' table view argument to determine which table view is sending the message:

```
- (NSInteger)tableView:(UITableView *)tableView
numberOfRowsInSection:(NSInteger)section {

    if (tableView == self.tableView) {
        return ...;
    }
    // If necessary (if self is the data source for other table views),
    // check whether tableView is searchController.searchResultsTableView.
    return ...;
}
```

**Important:** Any given view controller or search bar can only be associated with a single search display controller at a time. If a search display controller is destroyed (for example, in response to a memory warning), then you can create a new one and associate it with the original view controller or search bar.

# Tasks

## Initialization

`– initWithSearchBar:contentsController:` (page 9)
Returns a display controller initialized with the given search bar and contents controller.

## Displaying the Search Interface

`active` (page 7) *property*
The visibility state of the search interface.

`– setActive:animated:` (page 9)
Displays or hides the search interface, optionally with animation.

## Configuration

`delegate` (page 7) *property*
The controller's delegate.

`searchBar` (page 8) *property*
The search bar. (read-only)

`searchContentsController` (page 8) *property*
The view controller that manages the contents being searched. (read-only)

`searchResultsTableView` (page 9) *property*
The table view in which the search results are displayed. (read-only)

`searchResultsDataSource` (page 8) *property*
The data source for the table view in which the search results are displayed.

`searchResultsDelegate` (page 8) *property*
The delegate for the table view in which the search results are displayed.

# Properties

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

## active

The visibility state of the search interface.

```
@property(nonatomic, getter=isActive) BOOL active
```

**Discussion**
The default value is `NO`.

If you set this value directly, any change is performed without animation. Use `setActive:animated:` (page 9) if a change in state should be animated.

When the user focus in the search field of a managed search bar, the search display controller automatically displays the search interface. You can use this property to force the search interface to appear.

**Availability**
Available in iOS 3.0 and later.

**Declared In**
`UISearchDisplayController.h`

## delegate

The controller’s delegate.

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

**Availability**
Available in iOS 3.0 and later.

**Declared In**
`UISearchDisplayController.h`

## searchBar

The search bar. (read-only)

```
@property(nonatomic, readonly) UISearchBar *searchBar
```

**Availability**
Available in iOS 3.0 and later.

**Declared In**
`UISearchDisplayController.h`

## searchContentsController

The view controller that manages the contents being searched. (read-only)

```
@property(nonatomic, readonly) UIViewController *searchContentsController
```

**Discussion**
This is typically an instance of `UITableViewController`.

**Availability**
Available in iOS 3.0 and later.

**Declared In**
`UISearchDisplayController.h`

## searchResultsDataSource

The data source for the table view in which the search results are displayed.

```
@property(nonatomic, assign) id<UITableViewDataSource> searchResultsDataSource
```

**Discussion**
The default is `nil`.

**Availability**
Available in iOS 3.0 and later.

**Declared In**
`UISearchDisplayController.h`

## searchResultsDelegate

The delegate for the table view in which the search results are displayed.

```
@property(nonatomic, assign) id<UITableViewDelegate> searchResultsDelegate
```

**Discussion**
The default is `nil`.

**Availability**
Available in iOS 3.0 and later.

**Declared In**
`UISearchDisplayController.h`

### searchResultsTableView

The table view in which the search results are displayed. (read-only)

```
@property(nonatomic, readonly) UITableView *searchResultsTableView
```

**Discussion**
This method creates a new table view if one does not already exist.

**Availability**
Available in iOS 3.0 and later.

**Declared In**
`UISearchDisplayController.h`

## Instance Methods

### initWithSearchBar:contentsController:

Returns a display controller initialized with the given search bar and contents controller.

```
- (id)initWithSearchBar:(UISearchBar *)searchBar contentsController:(UIViewController
     *)viewController
```

**Parameters**

*searchBar*

A search bar.

The search bar must not currently be associated with another search display controller.

*viewController*

The view controller that manages display of the original contents that are to be searched.

The view controller must not currently be associated with another search display controller.

**Return Value**
A display controller initialized with the given search bar and contents controller.

**Availability**
Available in iOS 3.0 and later.

**Declared In**
`UISearchDisplayController.h`

### setActive:animated:

Displays or hides the search interface, optionally with animation.

```
- (void)setActive:(BOOL)visible animated:(BOOL)animated
```

### Parameters

*visible*

YES to display the search interface if it is not already displayed; NO to hide the search interface if it is currently displayed.

*animated;*

YES to use animation for a change in visible state, otherwise NO.

### Discussion

When the user focus in the search field of a managed search bar, the search display controller automatically displays the search interface. You can use this method to force the search interface to appear.

### Availability

Available in iOS 3.0 and later.

### Declared In

`UISearchDisplayController.h`

# Document Revision History

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

| Date | Notes |
|---|---|
| 2009-08-13 | Corrected the declaration of searchContentsController. |
| 2009-04-21 | Added note that a controller has a to-one relationship with a search bar and a view controller. |
| 2009-02-09 | New document that describes a controller that manages display of a search interface. |