



# iOS Development

iOS Development Tips and Tricks

# Method Documentation

// MARK: - Table view delegate

```
C @interface ProfileTableViewController()
C @implementation ProfileTableViewController
P @synthesize profilesArray, fugitiveViewController;
M -initWithStyle:
M -viewDidLoad
M -viewWillAppear:
M -viewDidUnload
M -shouldAutorotateToInterfaceOrientation:
```

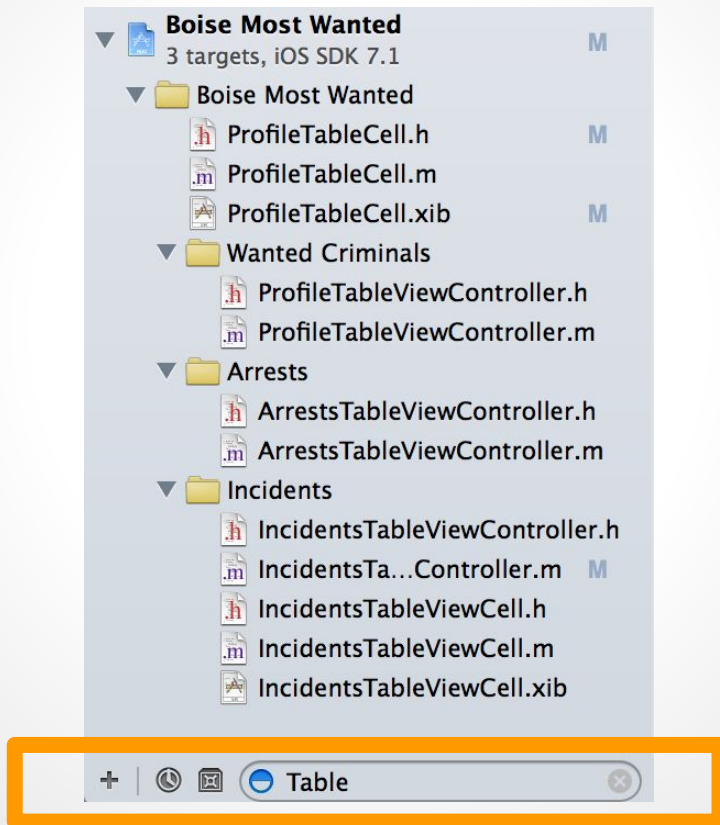
## Table view data source

```
M -numberOfSectionsInTableView:
M -tableView:numberOfRowsInSection:
M -tableView:cellForRowAtIndexPath:
```

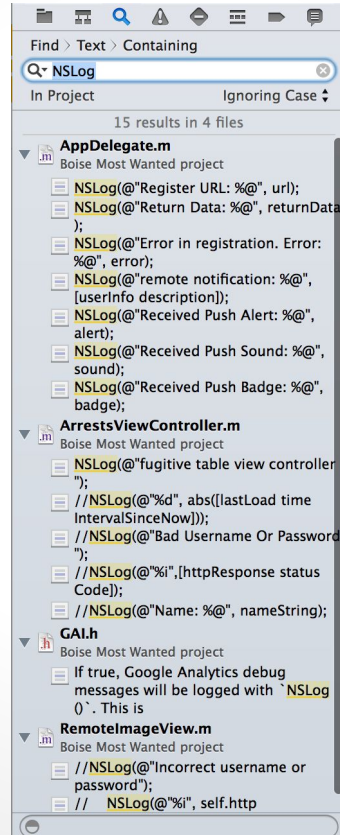
## Table view delegate

```
M -tableView:heightForRowAtIndexPath:
M -tableView:didSelectRowAtIndexPath:
```

# File Name Search



# Search Within Files Command-4



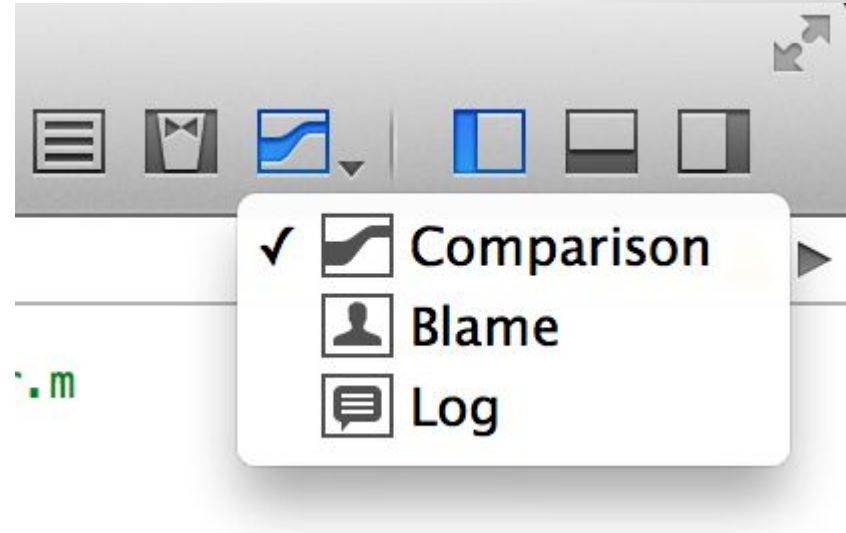
# Source Control

## Comparison -

Compare revisions

**Blame** - Find out who's to blame

**Log** - See the revision comments



# Refactor - Rename



# Refactor - Extract

Extract to

☒ New method    ☐ New function

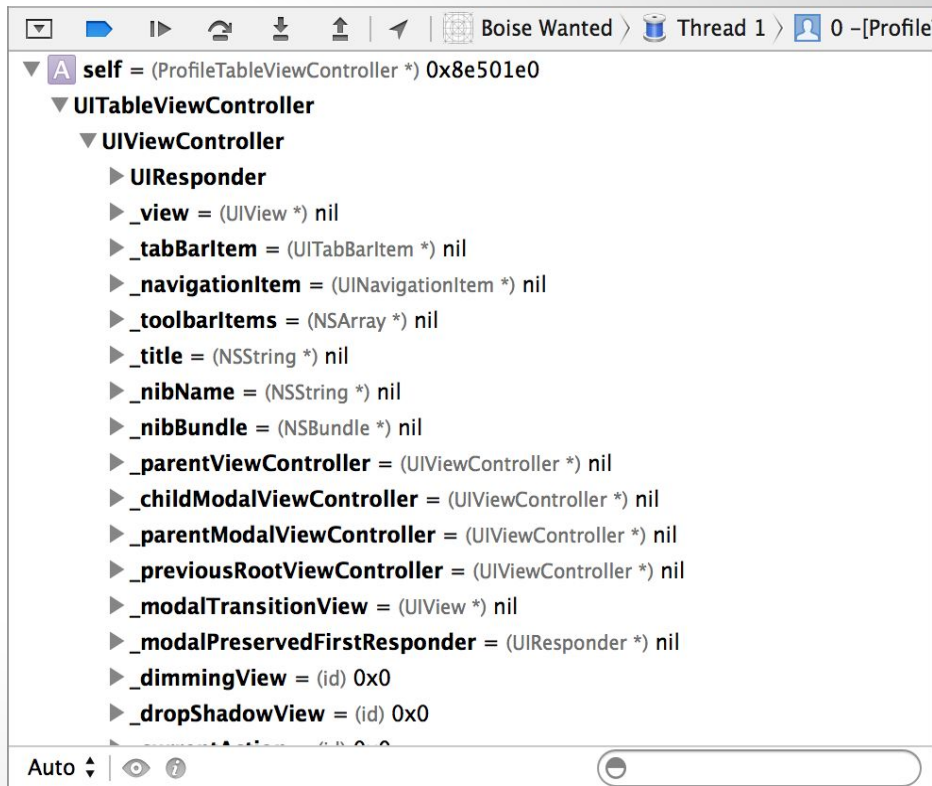
25	- (void)extracted_method	1 ▾	25
26	{		26
27	CGRect screenFrame = [[UIScreen mainScreen] bounds];		27
28	UILabel *centeredLabel = [[UILabel alloc] initWithFrame: CGRectMake(0,	2 ▾	28
29	centeredLabel.text = @"Label Text";		29
30	centeredLabel.backgroundColor = [UIColor clearColor];		30
31	centeredLabel.font = [UIFont fontWithName:@"Helvetica" size: 14];		31
32	[self.window addSubview: centeredLabel];		32

# Debug Area

Filter Variables

Quick Look

Print Description



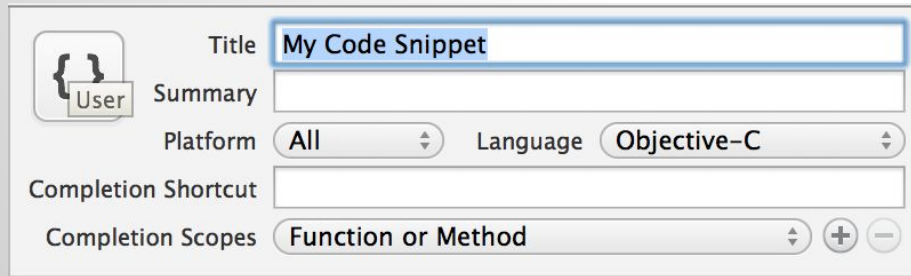


# Debug Area

Demo

# Code Snippets

Add your own, commonly used snippets of code.  
(Right click highlighted code to add)



A screenshot of the 'Code Snippet Editor' dialog box. It features a title bar with a document icon and a 'User' label. The main area contains a 'Title' field with the text 'My Code Snippet', a 'Summary' field, a 'Platform' dropdown menu set to 'All', and a 'Language' dropdown menu set to 'Objective-C'. At the bottom, there is a 'Completion Shortcut' field and a 'Completion Scopes' dropdown menu set to 'Function or Method'.

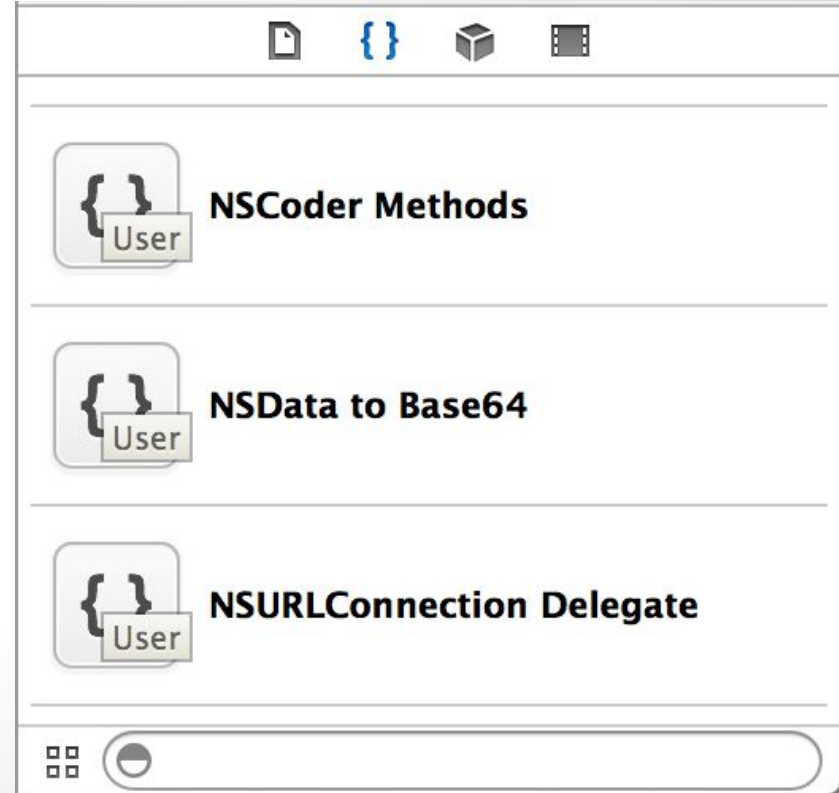
Title: My Code Snippet

Summary:

Platform: All Language: Objective-C

Completion Shortcut:

Completion Scopes: Function or Method



# Quick Help - Option-Click

```
static var persistentContainer: NSPersistentContainer = {
```

```
/*
```

```
The persistentContainer property  
creates and manages the persistent  
application container. If the  
error condition is not nil, the  
error condition is nil.
```

```
*/
```

```
let container = NSPersistentContainer(name: "MyApp")  
container.loadPersistentStores(completionHandler: { (storeDescription, error) in
```

```
    if let error = error {  
        fatalError("Unrecognized error: \(error)")  
    }  
    // Return the implementation here. Use fatalError() to make the error app  
    // fatalError() causes the application to generate a crash log and terminate the application. This function is a shipping application, although it may be useful for development.
```

Declaration `class NSPersistentContainer : NSObject`

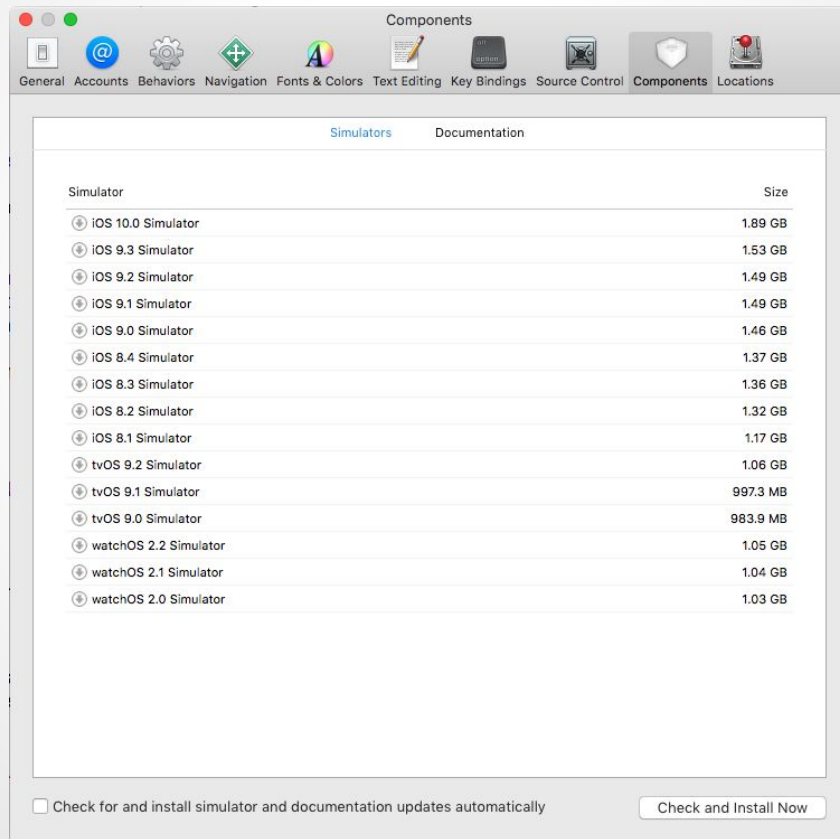
Description A container that encapsulates the Core Data stack in your application. NSPersistentContainer simplifies the creation and management of the Core Data stack by handling the creation of the `NSManagedObjectModel`, `NSPersistentStoreCoordinator`, and the `NSManagedObjectContext`.

Availability iOS (10.0 and later), macOS (10.12 and later), tvOS (10.0 and later), watchOS (3.0 and later)

Declared In `Core Data`

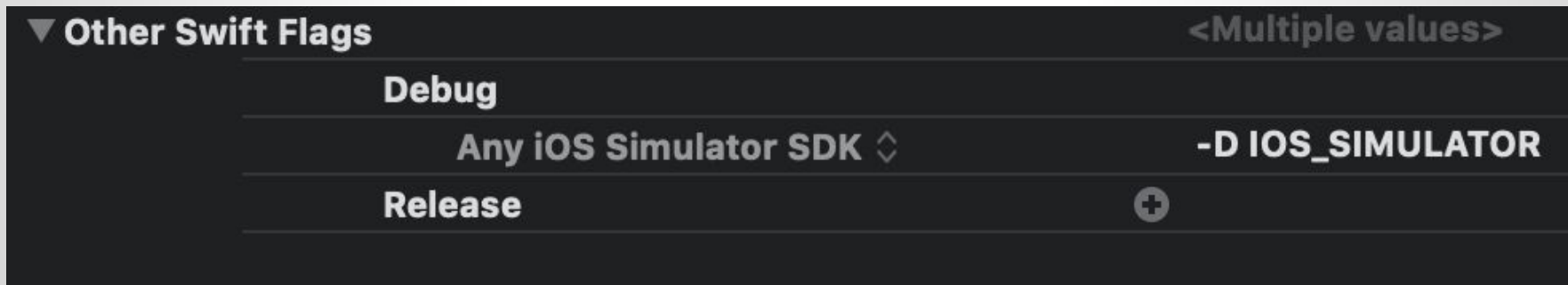
More [Class Reference](#)

# Adding Older Simulators - CMD-,



# Compiler Macros and Directives

Useful for including or excluding code during compilation



# Compiler Macros and Directives - Example

```
#if PROD
let SERVER_LOCATION = "http://www.production.com"

#elseif QA
let SERVER_LOCATION = "http://test.production.com/"

#elseif DEV
let SERVER_LOCATION = "http://test.zstudiolabs.com"
#endif
```