

# Developer Tools Kickoff

Matthew Firlik

Director, Development Technologies





# Xcode 4.0

March 2011



# Apple LLVM Compiler

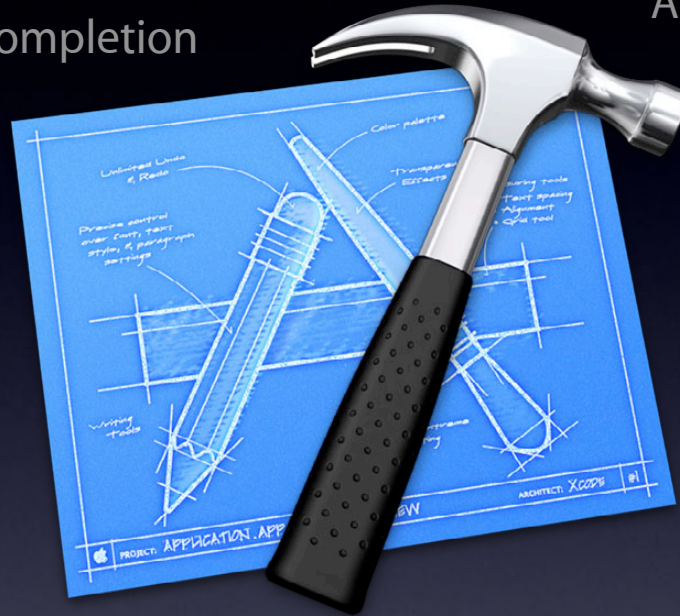
Schemes

Automatic Provisioning

Archives

Code Completion

## Single Window



## Interface Builder

Subversion Git

Branching & Merging

## Version Editor

Tabs

Unit Testing

Behaviors Quick Help

Live Issues

## Assistant Editor

Build Settings View

C++ in LLVM

Workspaces

One-Click Filtering

## LLDB

App Validation

Code Snippets

## Fix-It

Blame

Jump Bar

Static Analysis

## Instruments

Archives

App Store Submission



Auto Layout for OS X

Custom Behaviors

Preprocess/Assembly Assistant



iOS Storyboards

Automatic Reference Counting

OpenGL ES Debugger



Developer ID

Self-Contained Packaging

Separate Tools Packages













Graphics Tools



Audio Tools



Hardware Tools



Accessibility Tools



Auxiliary Tools

# Command Line Tools





# Xcode

Everything you need to create great apps for Mac, iPhone, and iPad.

Free



## Xcode

Xcode provides everything developers need to create great applications for Mac, iPhone, and iPad. Xcode 4 has been streamlined to help you write better apps. It has unified user interface design, coding, testing, and debugging all within a single window. The Xcode IDE analyzes the details of your project to identify mistakes in both syntax and logic, it can even help fix your code for you.

...

### What's New in Version 4.3.3

Xcode is now distributed as an application, rather than as an installer. This change enables Xcode to be updated directly from the Mac App Store.

...

...More

- [Apple Web Site](#) >
- [Xcode Support](#) >
- [App License Agreement](#) >

### Information

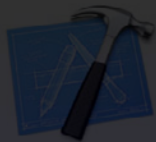
Category: Developer Tools  
 Version: 4.3.3  
 Price: Free  
 Size: 1.43 GB  
 Language: English  
 Seller: Apple Inc.  
 © 2012 Apple Inc.

Rated 4+  
 Requirements:  
 OS X 10.7.3 or later

### More by Apple

- OS X Lion**  
 Productivity  
 ★★★★★
- Final Cut Pro**  
 Video  
 ★★★★★
- Pages**  
 Productivity  
 ★★★★★
- iPhoto**  
 Photography  
 ★★★★★





# Xcode

Everything you need to create great apps for Mac, iPhone, and iPad.

Free



## Xcode

Xcode provides everything developers need to create great applications for Mac, iPhone, and iPad. Xcode 4 has been streamlined to help you write better apps. It has unified user interface design, coding, testing, and debugging all within a single window. The Xcode IDE analyzes the details of your project to identify mistakes in both syntax and logic, it can even help fix your code for you.

## What's New In Version 4.3.3

Xcode is now distributed as an application, rather than as an installer. This change enables Xcode to be updated directly from the Mac App Store.

...More

[Apple Web Site](#)

[Xcode Support](#)

[App License Agreement](#)

...More

## Information

Category: Developer Tools  
Version: 4.3.3  
Price: Free  
Size: 1.43 GB  
Language: English  
Seller: Apple Inc.  
© 2012 Apple Inc.

Rated 4+

Requirements:  
OS X 10.7.3 or later

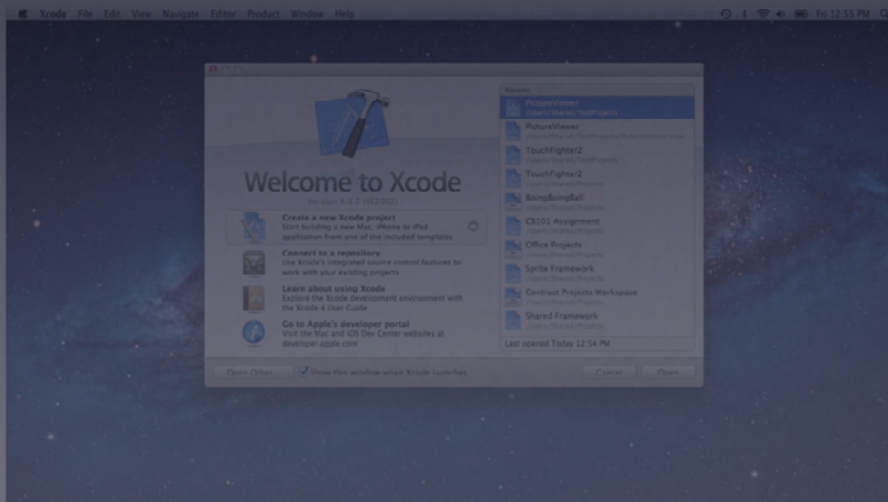
## More by Apple

[OS X Lion](#)  
Productivity  
★★★★

[Final Cut Pro](#)  
Video  
★★★★

[Pages](#)  
Productivity  
★★★★

[iPhoto](#)  
Photography  
★★★★





The image shows a screenshot of the Xcode app page on the Mac App Store. A large, semi-transparent blue arrow with a white outline points downwards from the top center towards the 'Free' price tag. The app page includes the Xcode logo, a description, a 'Free' price tag, and a 'What's New In Version 4.3.3' section. A 'Welcome to Xcode' dialog box is visible in the foreground, and the right sidebar shows app details and 'More by Apple' recommendations.

# Xcode

Everything you need to create great apps for Mac, iPhone, and iPad.

**Free**

## Xcode

Xcode provides everything developers need to create great applications for Mac, iPhone, and iPad. It has unified user interface design, coding, testing, and debugging all within a single window. It has been streamlined to help you write better apps. It analyzes the details of your project to identify mistakes in both syntax and logic, it can even help fix your code.

### What's New In Version 4.3.3

Xcode is now distributed as an application, rather than as an installer.

**Information**

Category: Developer Tools  
Version: 4.3.3  
Price: Free  
Size: 1.43 GB  
Language: English  
Seller: Apple Inc.  
© 2012 Apple Inc.

Rated 4+  
Requirements:  
OS X 10.7.3 or later

#### More by Apple

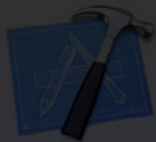
- OS X Lion Productivity
- Final Cut Pro Video
- Pages Productivity
- iPhoto Photography

**Welcome to Xcode**  
Version 4.3.3 (12E2002)

- Create a new Xcode project  
Start building a new Mac, iPhone or iPad application from one of the included templates.
- Connect to a repository  
Use Xcode's integrated source control features to work with your existing projects.
- Learn about using Xcode  
Explore the Xcode development environment with the Xcode 4 User Guide.
- Go to Apple's developer portal  
Visit the Mac and iOS Dev Center websites at developer.apple.com.

QuickStarts:  
TouchFlight2  
BonjourBall  
CS101 Assignment  
Office Projects  
Sprite Framework  
Contract Projects Workspace  
Shared Framework

Last opened Today 12:14 PM



# Xcode

Everything you need to create great apps for Mac, iPhone, and iPad.

Free

## Xcode

Xcode provides everything developers need to create great apps for Mac, iPhone, and iPad. It has unified user interface design, coding, testing, and debugging tools. It can even help you identify mistakes in both syntax and logic.

## What's New in Version 4.3.3

Xcode is now distributed as an application, rather than as a system component. This means you can install it on any Mac that supports OS X 10.7.3 or later.



write better apps. Select to identify

...More

- [Apple Web Site](#)
- [Xcode Support](#)
- [App License Agreement](#)

## Information

Category: Developer Tools  
 Version: 4.3.3  
 Price: Free  
 Size: 1.43 GB  
 Language: English  
 Seller: Apple Inc.  
 © 2012 Apple Inc.

Rated 4+  
 Requirements:  
 OS X 10.7.3 or later

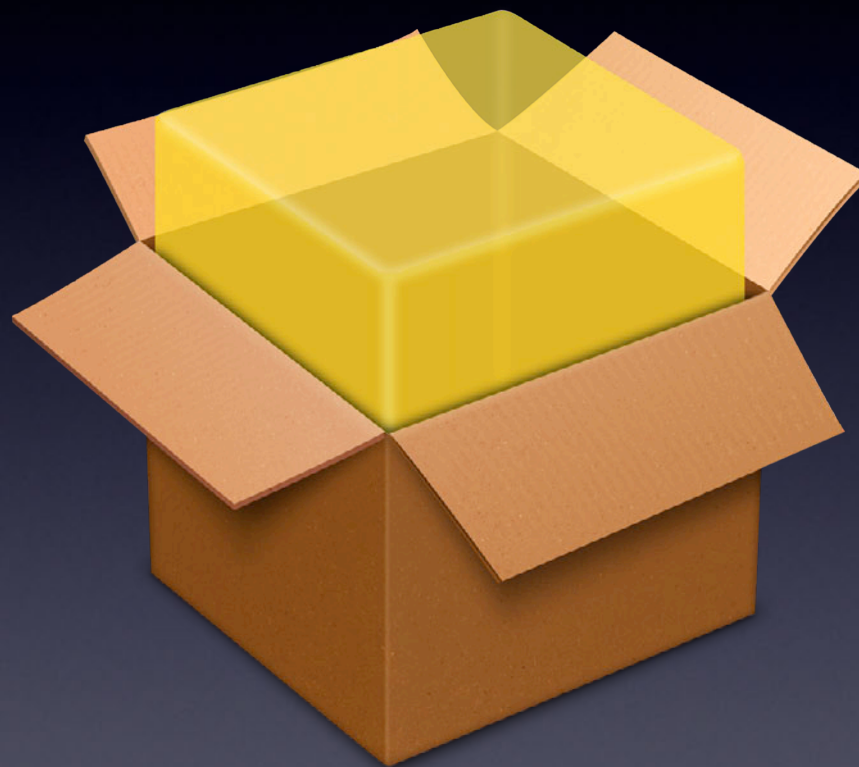
## More by Apple

- [OS X Lion Productivity](#) ★★★★★
- [Final Cut Pro Video](#) ★★★★★
- [Pages Productivity](#) ★★★★★
- [iPhoto Photography](#) ★★★★★



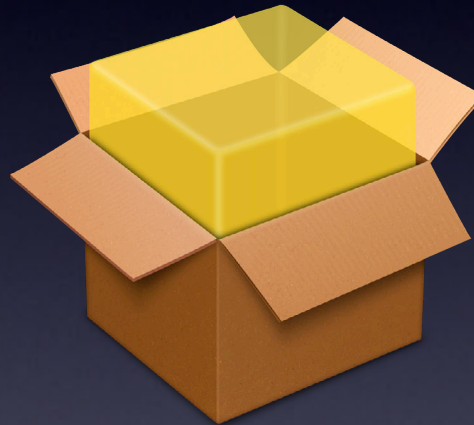
# Xcode 4.1

4GB



# Xcode 4.2

1.5GB

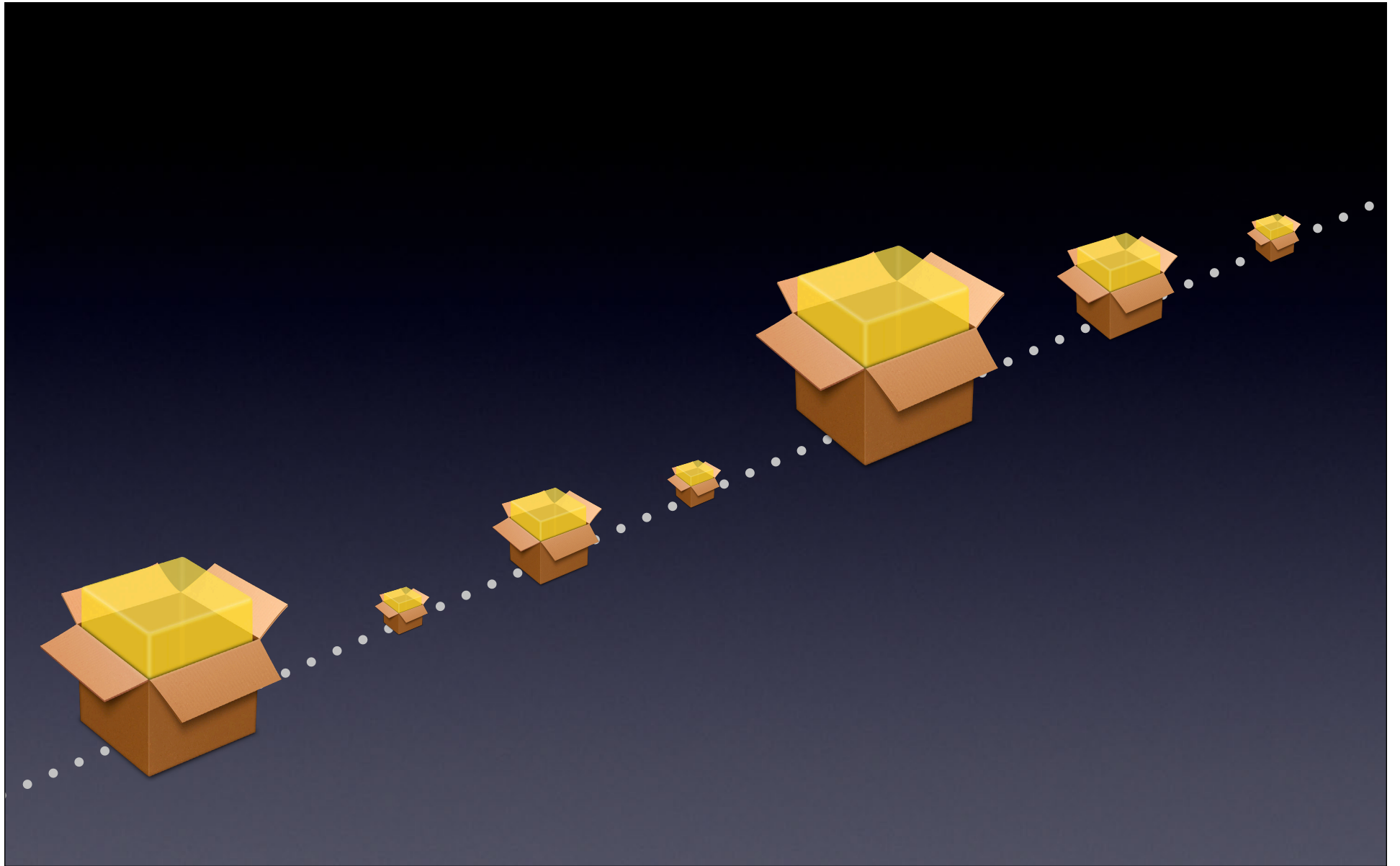


# Xcode 4.3.3

97MB













Xcode 4.4



Xcode 4.5

Language

Language

# Objective-C L@nguage

@interface

@class

@property

@property  
@synthesize



@property

@retain @release

# Automatic Reference Counting

@

@ . . . .

@literals

@*""* strings

@*""* strings



@

@

@ ""

@

@

strings

@ # numbers

@ {} dictionaries

@ "" strings

@ [] arrays

@ () expressions

# Demonstration

Mike Ferris

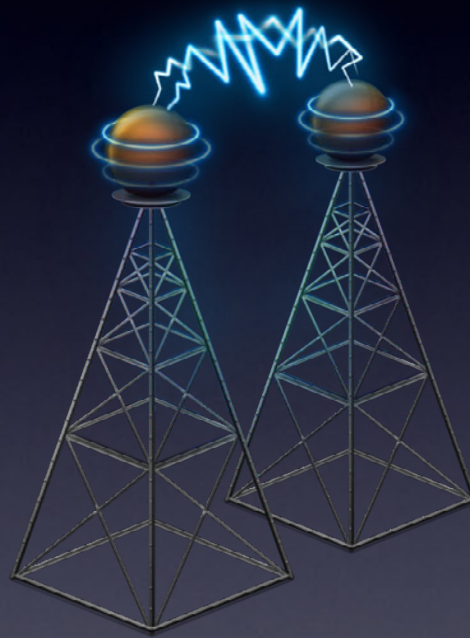
Senior Engineering Manager, Xcode

@literals

@property

@synthesize by default

# Migration Support for Garbage Collection



L@nguage

Edit



Edft



```
#import "SGTLandmarkRepository.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];
    NSIndexPath *selectedLandmarkPath = [[self tableView] indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance]
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell
{
    [cell setBackgroundColor:[UIColor clearColor]];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
    return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}
```

## Code Completion with Quick Help

```
#import "SGTLandmarkRepository.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];
    NSIndexPath *selectedLandmarkPath = [[self tableView] indexPathForSelectedRow];
    detailViewController.selectedLandmarkPath = selectedLandmarkPath;
    detailViewController.sharedRepository = sharedRepository;
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionSection:(NSInteger)section
    return [[SGTLandmarkRepository sharedRepository] numberOfLandmarks];
}

@end
```

C NSIndexPath

C NSIndexSet

The NSIndexPath class represents the path to a specific node in a tree of nested array collections. This path is known as an index path. [More...](#)

# Code Completion with Quick Help



```
#import "SGTLandmarkRepository.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];
    NSIndexPath *selectedLandmarkPath = [[self tableView] indexPathForSelectedRow];
    detailViewController.selectedLandmarkPath = selectedLandmarkPath;
    [detailViewController.tableView reloadData];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionSection:(NSInteger)section
    return [[SGTLandmarkRepository sharedRepository] numberOfLandmarks];
}

@end
```

C NSIndexPath

C NSIndexPath

The NSIndexPath class represents an immutable collection of unique unsigned integers, known as indexes because of the way they are used. This collection is referred to as an index set. [More...](#)

## Code Completion with Quick Help

```
#import "SGTLandmarkRepository.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];
    NSIndexPath *selectedIndexPath = [[self tableView] indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance]
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell
{
    [cell setBackgroundColor:[UIColor clearColor]];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
    return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}
```

## Code Completion with Quick Help

```
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];
    NSIndexPath *selectedIndexPath = [[self tableView] indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance]
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell
    [cell setBackgroundColor:[UIColor clearColor]];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionInSection:(NSInteger)section
    return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:
```

## Improved Quick Help



```
- (void) prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
    [super prepareForSegue:segue sender:sender],
    SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController]
    NSIndexPath *selectedIndexPath = [[self tableView] indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance]
    [detailViewController setLandmark:selectedLandmark];
}

- (void) tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell
    [cell setBackgroundColor:[UIColor clearColor]];
}

- (NSInteger) tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
    return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}

- (UITableViewCell *) tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
```

## Improved Quick Help

alt

option

?

```
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender  
    [super prepareForSegue:segue sender:sender];  
    SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];  
    NSIndexPath *selectedIndexPath = [[self tableView] indexPathForSelectedRow];  
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance] landmarkAtIndex:selectedIndexPath];  
    [detailViewController setLandmark:selectedLandmark];  
}  
  
- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell  
    [cell setBackgroundColor:[UIColor clearColor]];  
}  
  
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionInSection:(NSInteger)section  
    return [[[SGTLandmarkRepository sharedInstance] landmarks] count];  
}  
  
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
```

## Improved Quick Help

alt

option

```
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender  
    [super prepareForSegue:segue sender:sender];  
    SGT...  
    NSI...  
    SGT...  
    [de...  
}  
  
- (void...  
    [ce...  
}  
  
- (NSI...  
    ret...  
}  
  
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath
```

**Declaration** - (void) prepareForSegue:(UIStoryboardSegue \*)segue sender:(id)sender

**Abstract** Notifies the view controller that a segue is about to be performed.

**Parameters**

segue	The segue object containing information about the view controllers involved in the segue.
sender	The object that initiated the segue. You might use this parameter to perform different actions based on which control (or other object) initiated the segue.

**Availability** iOS (5.0 and later)

**Declared** [UIViewController.h](#)

**Reference** [UIViewController Reference](#)

# Improved Quick Help

```

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
  [super prepareForSegue:segue sender:sender];
  SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];
  NSIndexPath *selectedLandmarkPath = [[self tableView] indexPathForSelectedRow];
  SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance]
  [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell
  [cell setBackgroundColor:[UIColor clearColor]];
  [cell setNeedsLayout];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
  return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:
  SGTLandmarkTableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:

```

## Trailing Whitespace Removal

```

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
  [super prepareForSegue:segue sender:sender];
  SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];
  NSIndexPath *selectedLandmarkPath = [[self tableView] indexPathForSelectedRow];
  SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance]
  [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell
  [cell setBackgroundColor:[UIColor clearColor]];
  [cell setNeedsLayout];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
  return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
  SGTLandmarkTableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:@"SGTLandmarkTableViewCell"];

```

## Trailing Whitespace Removal

```
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];
    NSIndexPath *selectedIndexPath = [[self tableView] indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance]
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell
    [cell setBackgroundColor:[UIColor clearColor]];
    [cell setNeedsLayout];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionInSection:(NSInteger)section
    return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:
```

## Trailing Whitespace Removal

```

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];
    NSIndexPath *selectedLandmarkPath = [[self tableView] indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance] landmarkAtIndex:selectedLandmarkPath];
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell
    [cell setBackgroundColor:[UIColor clearColor]];
    [cell setNeedsLayout];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionInSection:(NSInteger)section
    return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:

```

## Trailing Whitespace Removal

```

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
  [super prepareForSegue:segue sender:sender];
  SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];
  NSIndexPath *selectedLandmarkPath = [[self tableView] indexPathForSelectedRow];
  SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance]
  [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell
  [cell setBackgroundColor:[UIColor clearColor]];
  [cell setNeedsLayout];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionInSection:(NSInteger)section
  return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
  SGTLandmarkTableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:@"SGTLandmarkTableViewCell"];

```

## Trailing Whitespace Removal



Sightseer > View Controllers > SGTViewC

Replace

Replace All

Replace

Replace & Find

```
#import "SGTLandmarksViewController.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];
    NSIndexPath *selectedLandmarkPath = [[self tableView] indexPathForSelectedCell];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance]
    [detailViewController setLandmark:selectedLandmark];
}
}
```

# Replace in Selection

Sightseer > View Controllers > SGTViewC

Replace

selected

Replace All

Replace

Replace & Find

current

```
#import "SGTLandmarksViewController.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];
    NSIndexPath *selectedLandmarkPath = [[self tableView] indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance]
    [detailViewController setLandmark:selectedLandmark];
}
}
```

# Replace in Selection

Sightseer > View Controllers > SGTViewC

Replace

selected

Replace All

Replace

Replace & Find

current

```
#import "SGTLandmarksViewController.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];
    NSIndexPath *selectedLandmarkPath = [[self tableView] indexPathForSelectedCell];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance]
    [detailViewController setLandmark:selectedLandmark];
}
}
```

# Replace in Selection

alt  
option

Sightseer > View Controllers > SGTViewC

Replace

selected

Replace All   Replace   Replace & Find   current

```
#import "SGTLandmarksViewController.h"  
#import "SGTLandmark.h"  
  
@implementation SGTLandmarksViewController  
  
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender  
    [super prepareForSegue:segue sender:sender];  
    SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];  
    NSIndexPath *selectedLandmarkPath = [[self tableView] indexPathForSelectedRow];  
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance] landmarkAtIndex:selectedLandmarkPath.indexPathItemAtIndex:0];  
    [detailViewController setLandmark:selectedLandmark];  
}
```

# Replace in Selection

alt  
option

Sightseer > View Controllers > SGTViewC

Replace

selected

Replace All Replace Replace & Find current

```
#import "SGTLandmarksViewController.h"  
#import "SGTLandmark.h"  
  
@implementation SGTLandmarksViewController  
  
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender  
    [super prepareForSegue:segue sender:sender];  
    SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];  
    NSIndexPath *selectedLandmarkPath = [[self tableView] indexPathForSelectedRow];  
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance] landmarkAtIndex:selectedLandmarkPath];  
    [detailViewController setLandmark:selectedLandmark];  
}
```

# Replace in Selection

alt  
option

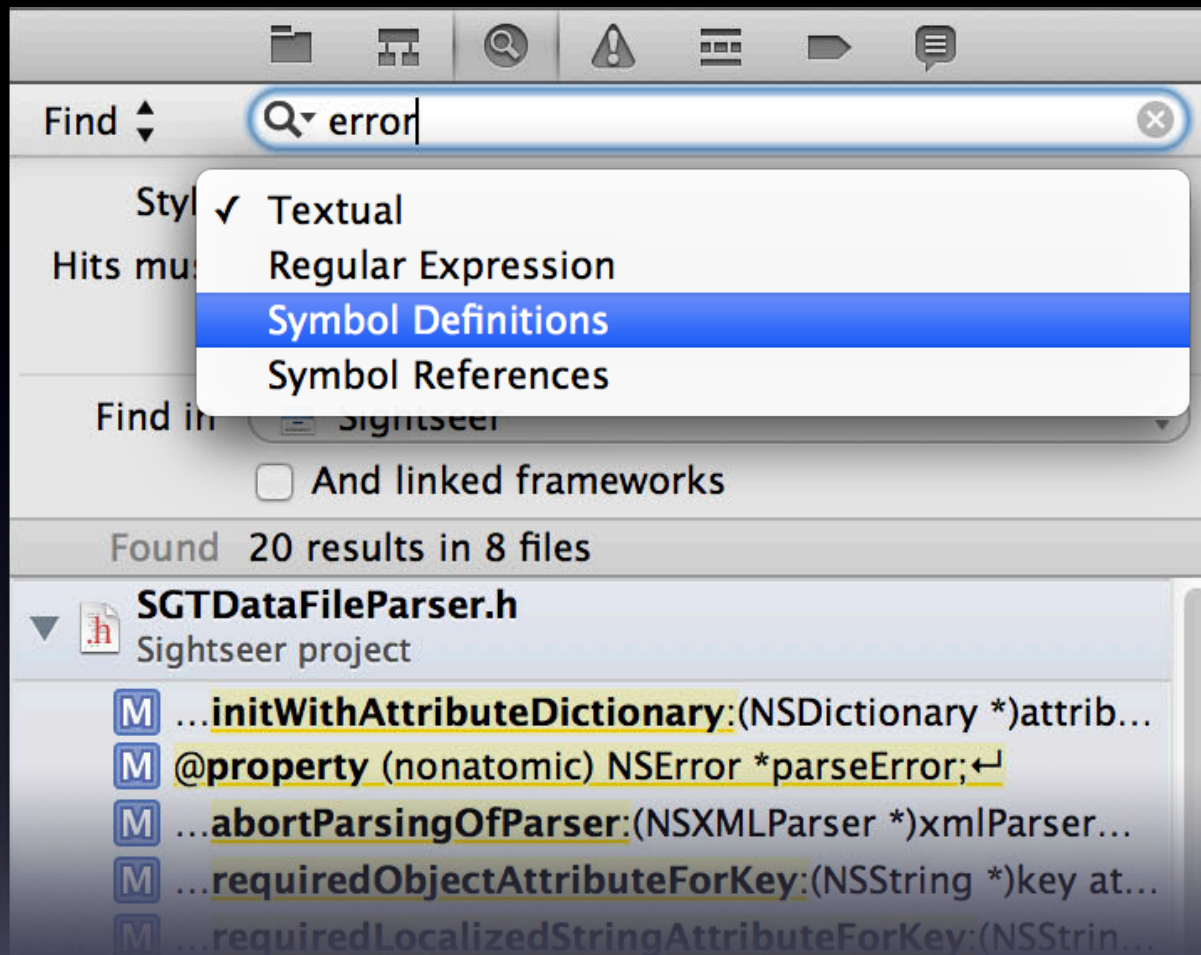
Sightseer > View Controllers > SGTViewC

Replace

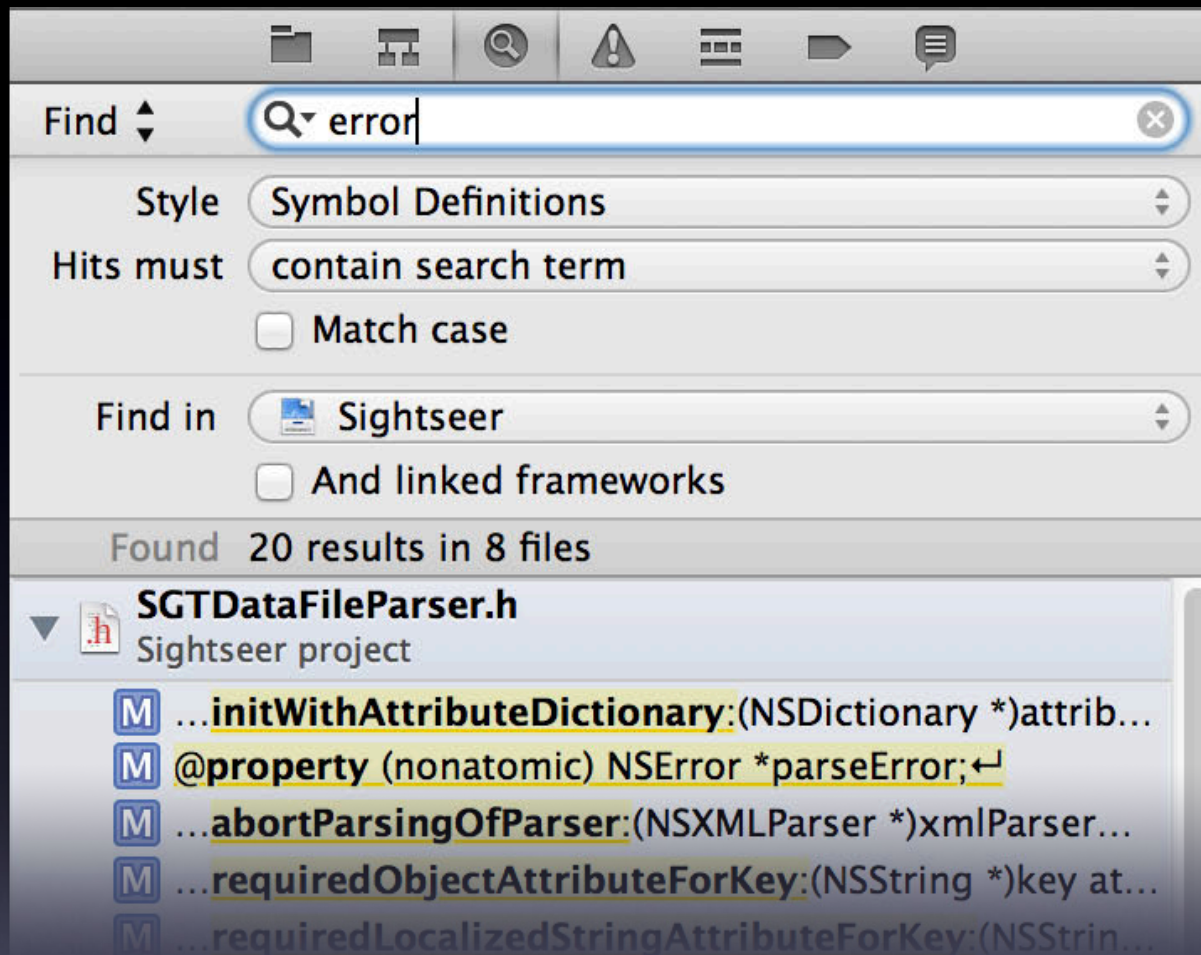
In Selection Replace Replace & Find current

```
#import "SGTLandmarksViewController.h"  
#import "SGTLandmark.h"  
  
@implementation SGTLandmarksViewController  
  
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender  
    [super prepareForSegue:segue sender:sender];  
    SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];  
    NSIndexPath *selectedLandmarkPath = [[self tableView] indexPathForSelectedObject];  
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance] landmarkAtIndex:selectedLandmarkPath];  
    [detailViewController setLandmark:selectedLandmark];  
}
```

# Replace in Selection



# Symbolic Searches



# Search Performance



---

grep

Xcode

A bar chart comparing the execution time of 'grep' and 'Xcode'. The 'grep' bar is significantly taller than the 'Xcode' bar, indicating a much longer execution time. The 'grep' bar is labeled with '19.3 seconds' at its top. The 'Xcode' bar is very short and lacks a numerical label. The chart has a dark blue background and a light blue horizontal axis.

Tool	Execution Time
grep	19.3 seconds
Xcode	~0.1 seconds

19.3 seconds

grep

Xcode

A bar chart comparing the execution times of two tools. The 'grep' tool is represented by a tall, dark grey bar with the value '19.3 seconds' written inside it. The 'Xcode' tool is represented by a shorter, blue bar with the value '4.6 seconds' written inside it. The bars are positioned above a horizontal baseline. The background is a dark gradient.

Tool	Execution Time
grep	19.3 seconds
Xcode	4.6 seconds

19.3 seconds

4.6 seconds

grep

Xcode



Callers (2) >



SGTViewController.m >

```
#import "SGTLandmarksViewController.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];
    NSIndexPath *selectedLandmarkPath = [[self tableView] indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance]
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell
{
    [cell setBackgroundColor:[UIColor clearColor]];
}
```

# Caller and Callee Assistant Editor

```

#import "SGTLandmarkRepository.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue
destinationViewContr
    NSIndexPath *selectedLandmarkPath = [[self tableView]
indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository
sharedInstance] landmarks
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell for
{cell setBackgroundColor:[UIColor clearColor];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionInSection:
(NSInteger)section
{ return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    SGTLandmarkTableViewCell *cell = [tableView
dequeueReusableCellWithIdentifier:@"La
    SGTLandmark *landmark = [[[SGTLandmarkRepository sharedInstance]
landmarks] object
    [[cell nameLabel] setText:[landmark localizedName]];
    return cell;
}

@end

```

```

#import "SGTLandmarkRepository.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue
destinationViewContr
    NSIndexPath *selectedLandmarkPath = [[self tableView]
indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository
sharedInstance] landmarks
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:
(UITableViewCell *)cell for
{cell setBackgroundColor:[UIColor clearColor];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionInSection:
(NSInteger)section
{ return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    SGTLandmarkTableViewCell *cell = [tableView
dequeueReusableCellWithIdentifier:@"La
    SGTLandmark *landmark = [[[SGTLandmarkRepository sharedInstance]
landmarks] object
    [[cell nameLabel] setText:[landmark localizedName]];
    return cell;
}

@end

```

# Selective Commit

```
#import "SGTLandmarkRepository.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue
destinationViewContr
    NSIndexPath *selectedLandmarkPath = [[self tableView]
indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository
sharedInstance] landmarks
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell for
{cell setBackgroundColor:[UIColor clearColor];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionInSection:
(NSInteger)section
{ return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    SGTLandmarkTableViewCell *cell = [tableView
dequeueReusableCellWithIdentifier:@"La
    SGTLandmark *landmark = [[[SGTLandmarkRepository sharedInstance]
landmarks] object
    [[cell nameLabel] setText:[landmark localizedName]];
    return cell;
}

@end
```



```
#import "SGTLandmarkRepository.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue
destinationViewContr
    NSIndexPath *selectedLandmarkPath = [[self tableView]
indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository
sharedInstance] landmarks
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:
(UITableViewCell *)cell for
{cell setBackgroundColor:[UIColor clearColor];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionInSection:
(NSInteger)section
{ return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    SGTLandmarkTableViewCell *cell = [tableView
dequeueReusableCellWithIdentifier:@"La
    SGTLandmark *landmark = [[[SGTLandmarkRepository sharedInstance]
landmarks] object
    [[cell nameLabel] setText:[landmark localizedName]];
    return cell;
}

@end
```

# Selective Commit

```
#import "SGTLandmarkRepository.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue
destinationViewContr
    NSIndexPath *selectedLandmarkPath = [[self tableView]
indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository
sharedInstance] landmarks
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell for
    [cell setBackgroundColor:[UIColor clearColor]];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
    return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath:
    SGTLandmarkTableViewCell *cell = [tableView
dequeueReusableCellWithIdentifier:@"La
    SGTLandmark *landmark = [[[SGTLandmarkRepository sharedInstance]
landmarks] objectAtIndex:indexPath.row];
    [cell nameLabel] setText:[landmark localizedName];
    return cell;
}

@end
```



1



```
#import "SGTLandmarkRepository.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue
destinationViewContr
    NSIndexPath *selectedLandmarkPath = [[self tableView]
indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository
sharedInstance] landmarks
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:
(UITableViewCell *)cell for
    [cell setBackgroundColor:[UIColor clearColor]];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
    return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath:
    SGTLandmarkTableViewCell *cell = [tableView
dequeueReusableCellWithIdentifier:@"La
    SGTLandmark *landmark = [[[SGTLandmarkRepository sharedInstance]
landmarks] objectAtIndex:indexPath.row];
    [cell nameLabel] setText:[landmark localizedName];
    return cell;
}

@end
```

# Selective Commit

```
#import "SGTLandmarkRepository.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue
destinationViewContr
    NSIndexPath *selectedLandmarkPath = [[self tableView]
indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository
sharedInstance] landmarks
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell for
*cell for
    [cell setBackgroundColor:[UIColor clearColor]];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
(NSInteger)section
    return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath:
(NSInteger)section
    SGTLandmarkTableViewCell *cell = [tableView
dequeueReusableCellWithIdentifier:@"La
    SGTLandmark *landmark = [[[SGTLandmarkRepository sharedInstance]
landmarks] objectAtIndex:indexPath.row];
    [cell nameLabel] setText:[landmark localizedName];
    return cell;
}

@end
```



1



Don't Commit

Discard Change

# Selective Commit



```
#import "SGTLandmarkRepository.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue
    destinationViewController
    NSIndexPath *selectedLandmarkPath = [[self tableView]
    indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository
    sharedInstance] landmarks
    detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell for
{cell setBackgroundColor:[UIColor clearColor];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:
{
    return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath {
    SGTLandmarkTableViewCell *cell = [tableView
    dequeueReusableCellWithIdentifier:@"La
    SGTLandmark *landmark = [[[SGTLandmarkRepository sharedInstance]
    landmarks] objectAtIndex:indexPath.row];
    [cell nameLabel] setText:[landmark localizedName];
    return cell;
}

@end
```

```
#import "SGTLandmarkRepository.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue
    destinationViewController
    NSIndexPath *selectedLandmarkPath = [[self tableView]
    indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository
    sharedInstance] landmarks
    detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:
(UITableViewCell *)cell for
{cell setBackgroundColor:[UIColor clearColor];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:
{
    return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView
```



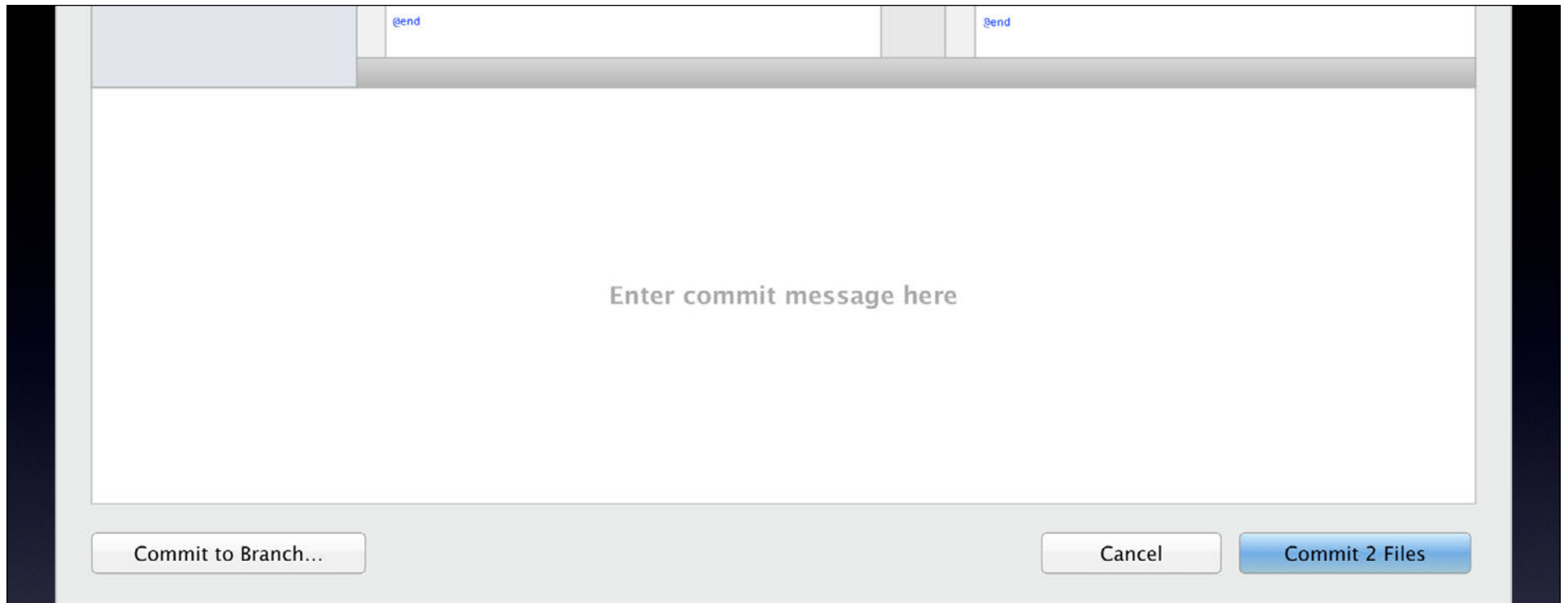
1



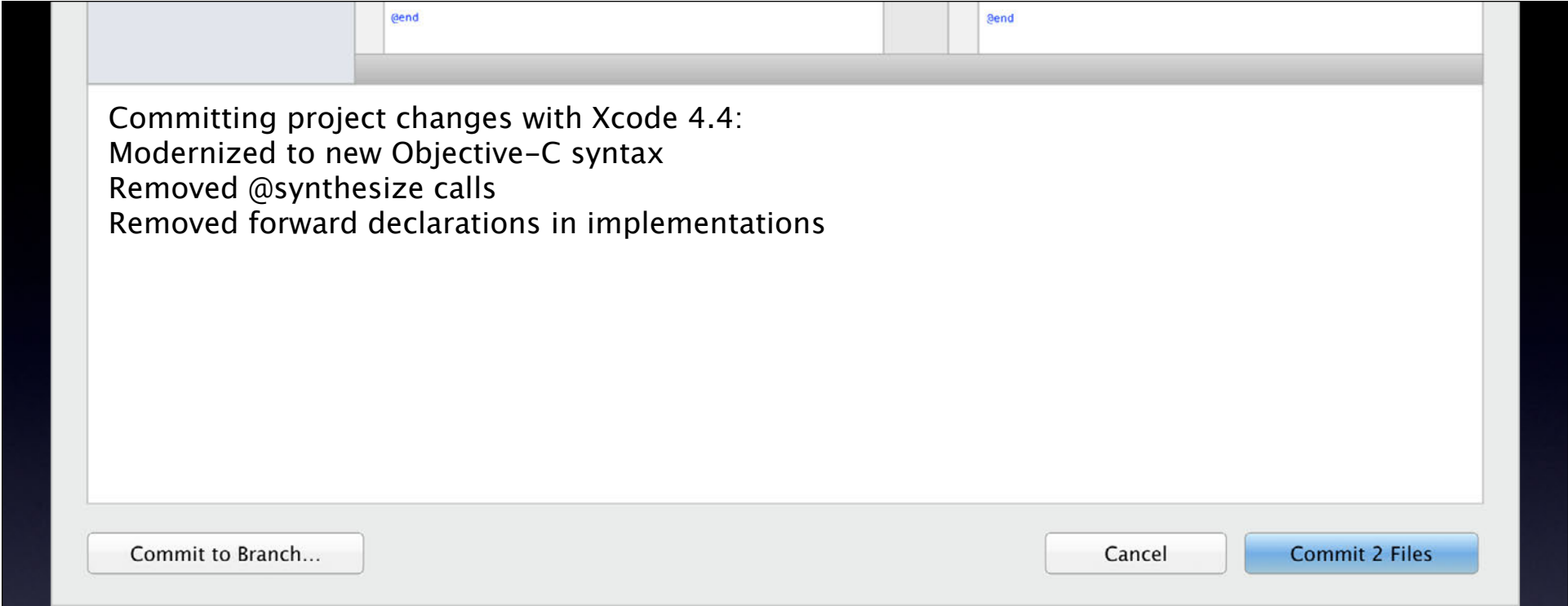
Don't Commit

Discard Change

# Selective Commit



# Source Control Branching



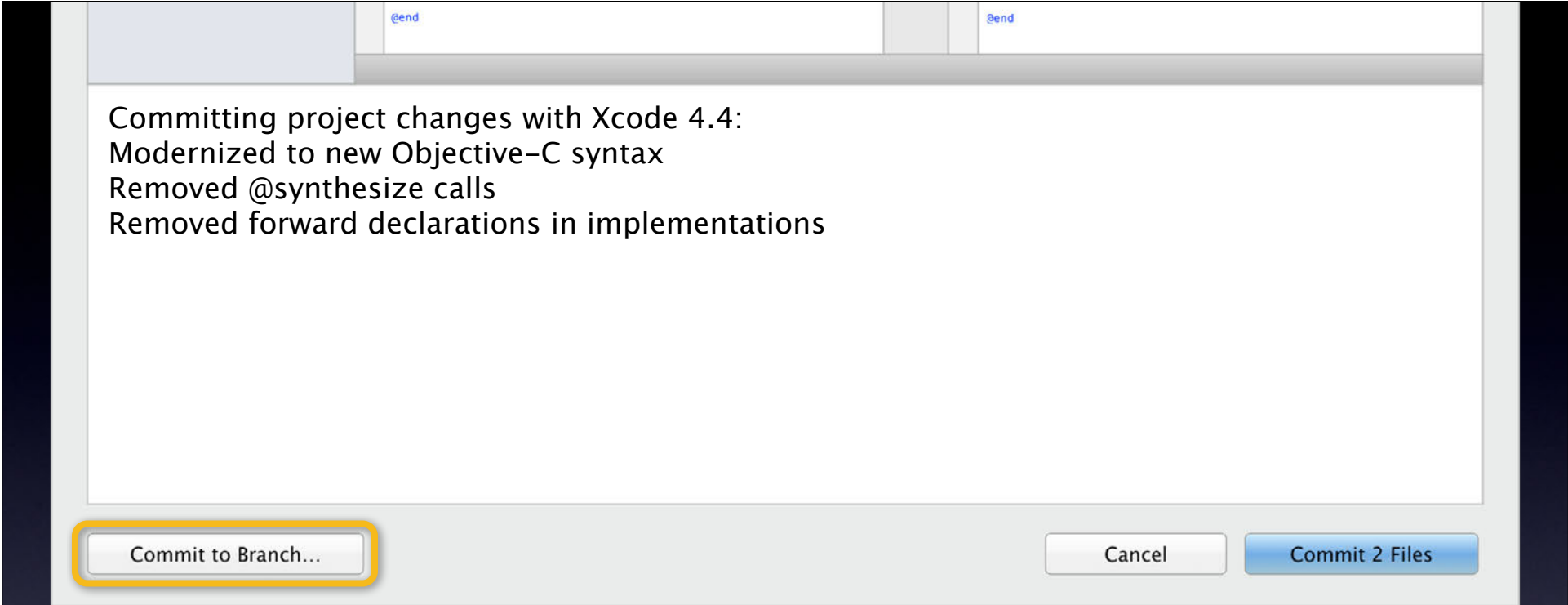
Committing project changes with Xcode 4.4:  
Modernized to new Objective-C syntax  
Removed @synthesize calls  
Removed forward declarations in implementations

Commit to Branch...

Cancel

Commit 2 Files

# Source Control Branching



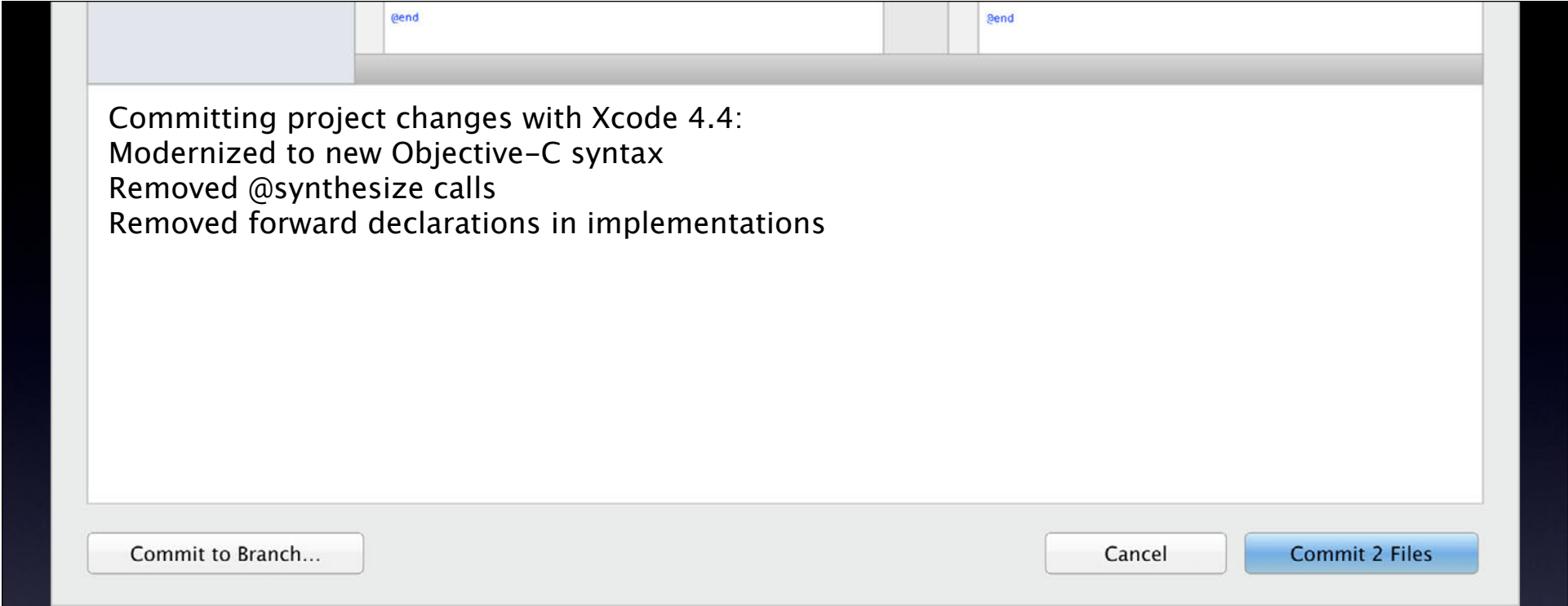
Committing project changes with Xcode 4.4:  
Modernized to new Objective-C syntax  
Removed @synthesize calls  
Removed forward declarations in implementations

Commit to Branch...

Cancel

Commit 2 Files

# Source Control Branching



Committing project changes with Xcode 4.4:  
Modernized to new Objective-C syntax  
Removed @synthesize calls  
Removed forward declarations in implementations

Commit to Branch...

Cancel

Commit 2 Files

# Source Control Branching

Committing awesome project changes with Xcode 4.4:  
Modernized to new Objective-C syntax  
Removed @synthesize calls  
Removed forward declarations in implementations  
Updated implementation to not use recursive lock

Commit to Branch...

Commit to Branch...

Cancel

Commit 2 Files

Source Control Branching

Edft

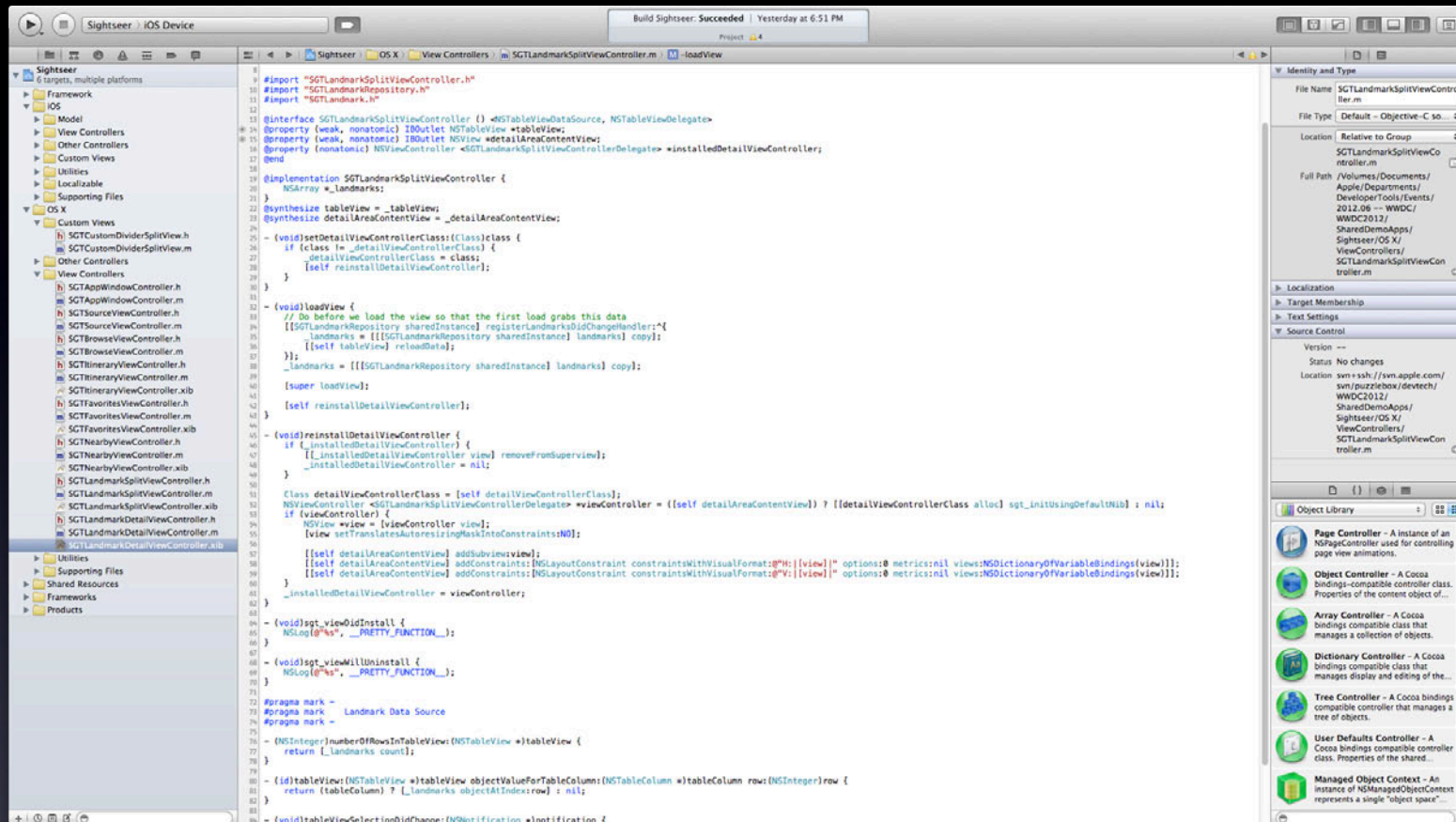
Design

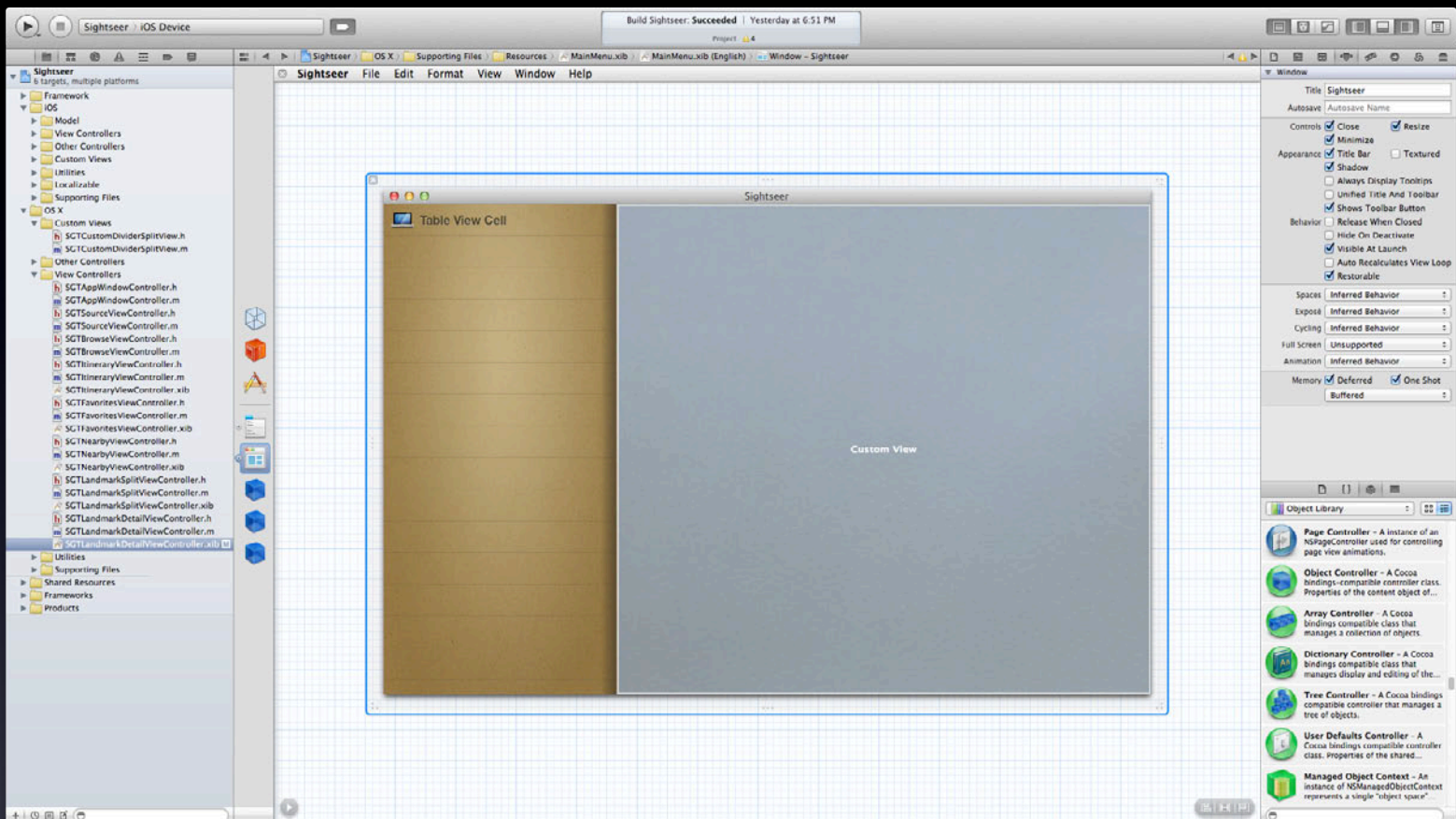


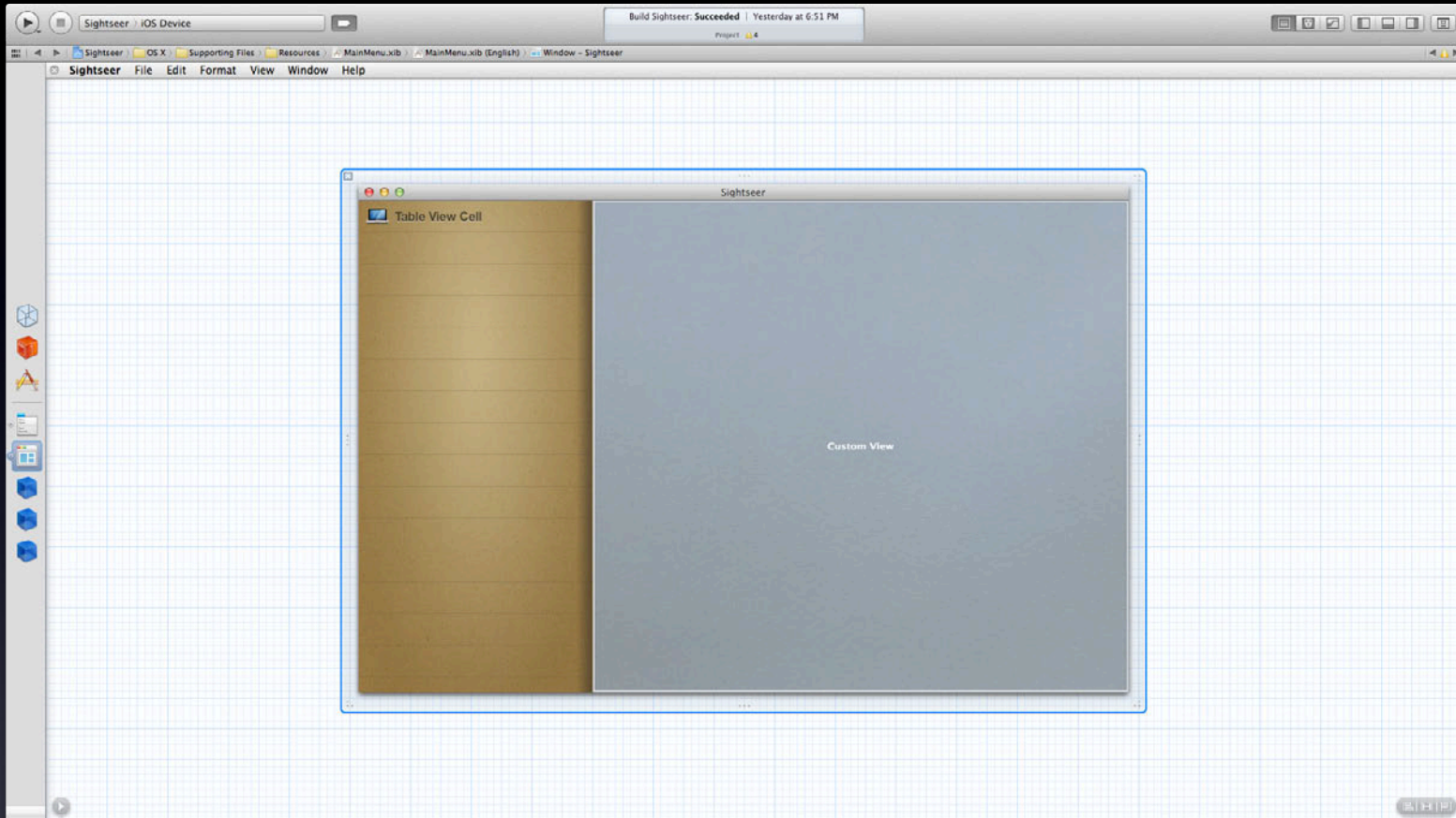
Design

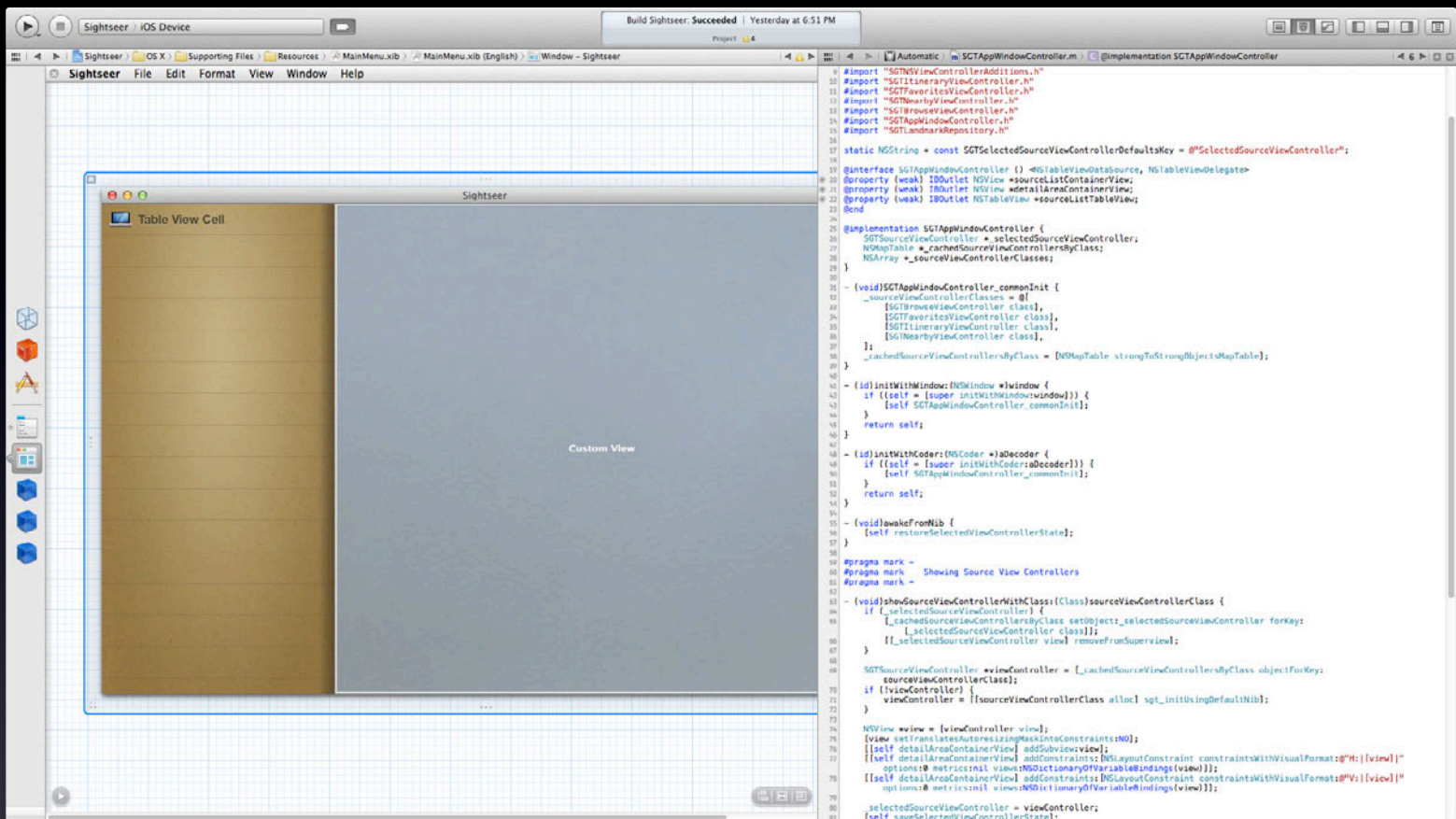
I

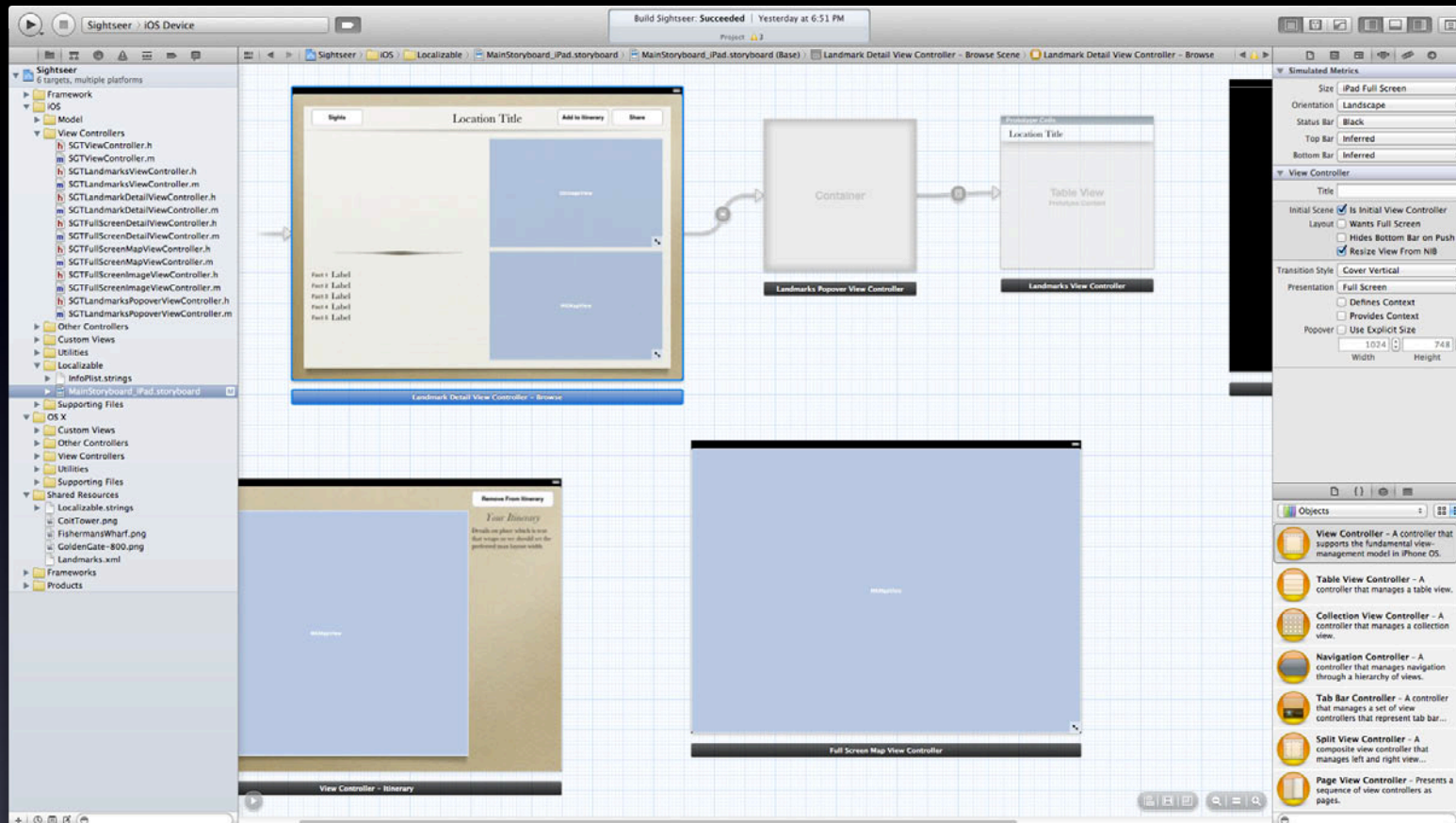
**I**ntegrated



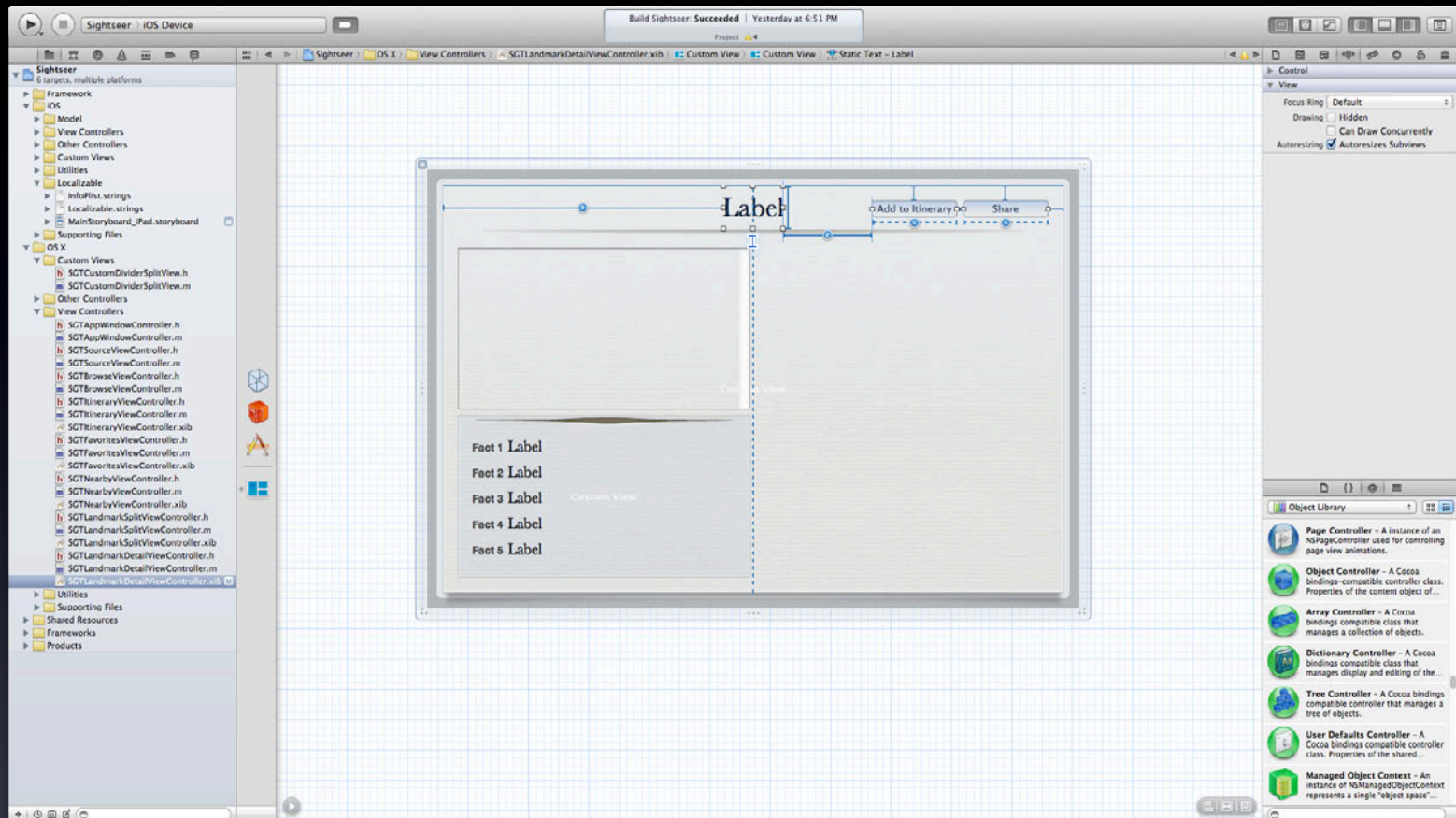


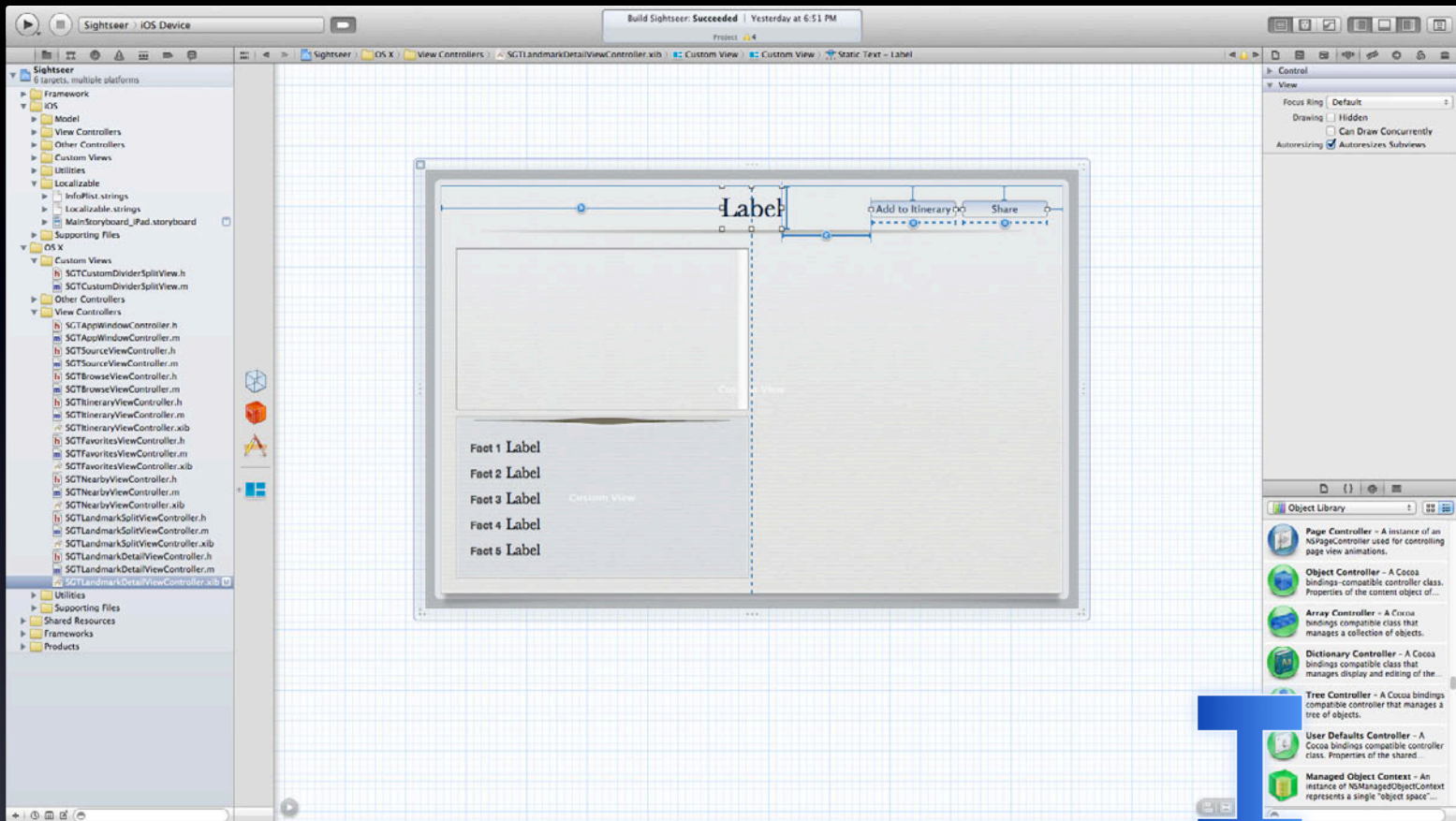




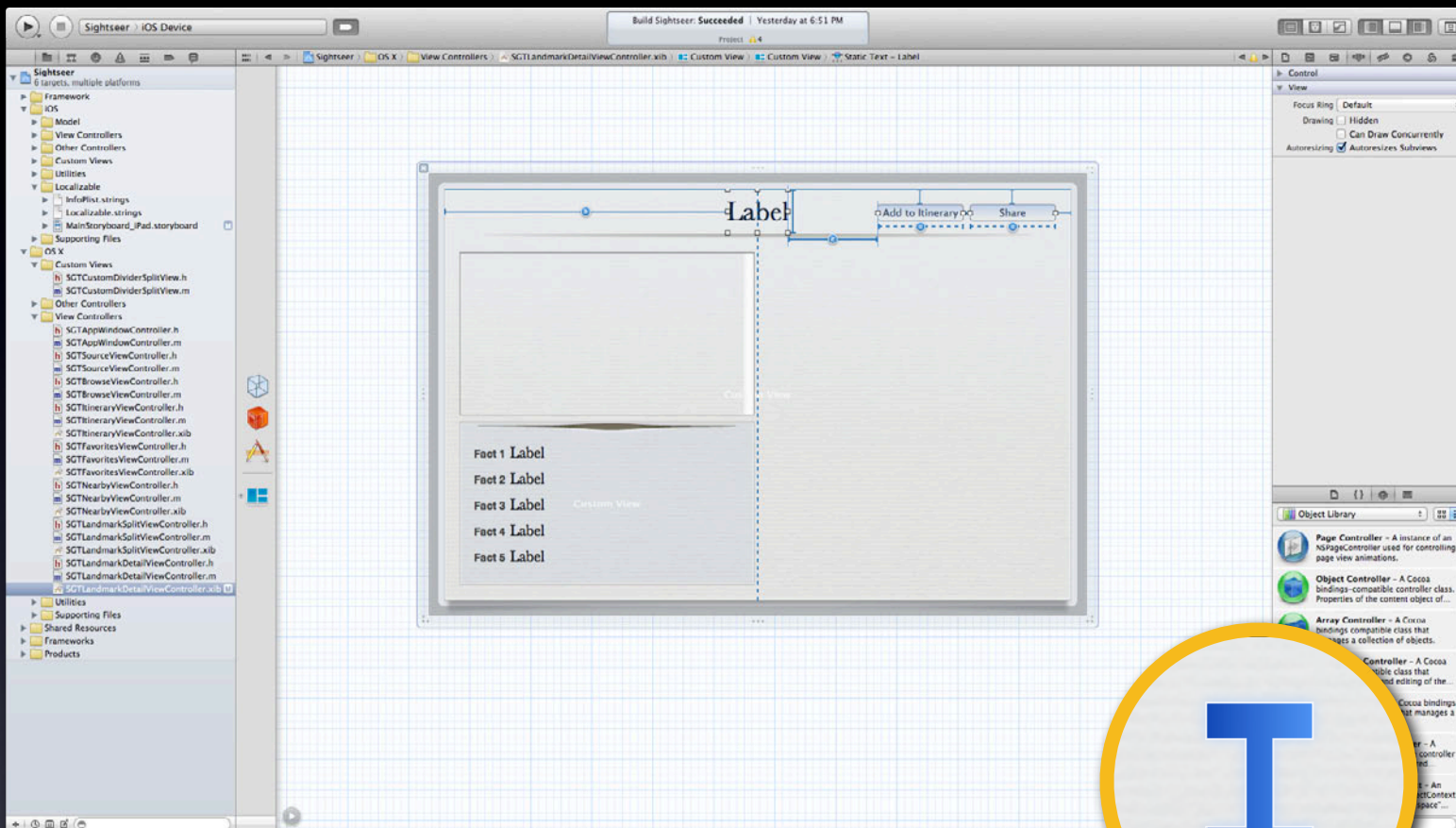


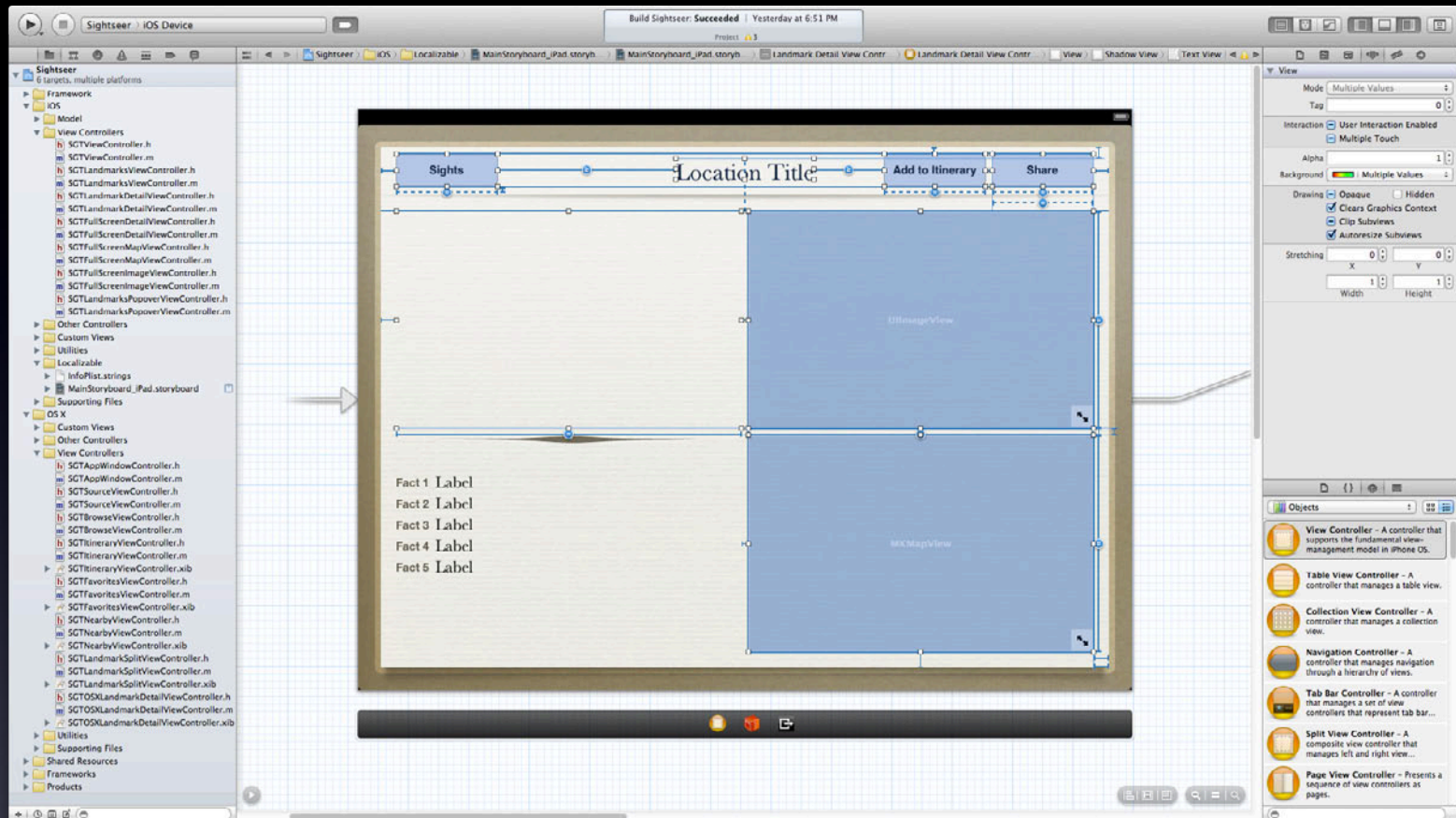






T






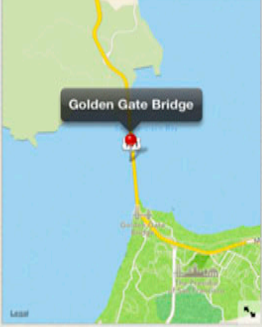
9:41 AM

Sights Golden Gate Bridge Add to Itinerary Share

Celebrating its 75th Anniversary in 2012, the Golden Gate Bridge is the most famous icon for the city of San Francisco. It spans the Golden Gate, the entrance to the San Francisco Bay, and connects San Francisco to its northern neighbors in Marin County. At the time of its opening, the Golden Gate Bridge was the longest suspension bridge in the world, coming in at a length of 4,200 feet. The renowned Golden color of the bridge is a color since named "International Orange," the color of the initial primer coat, later adopted as the permanent hue for the structure. Construction on the span started in 1933 and lasted four years, culminating in an opening "fiesta" starting on May 27, 1937. The design team of Joseph Strauss, Charles E. Ellis, and Louis Mooser was awarded with



LENGTH 8,981 ft (2,737.4 m),...  
WIDTH 90 ft (27.4 m)  
HEIGHT 746 ft (227.4 m)  
LONGEST SPAN 4,200 ft (1,280.2 m)



Pad

9:41 AM

Locations

# Golden Gate Bridge

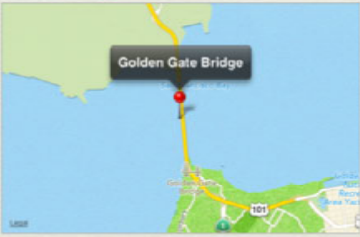
Checklist

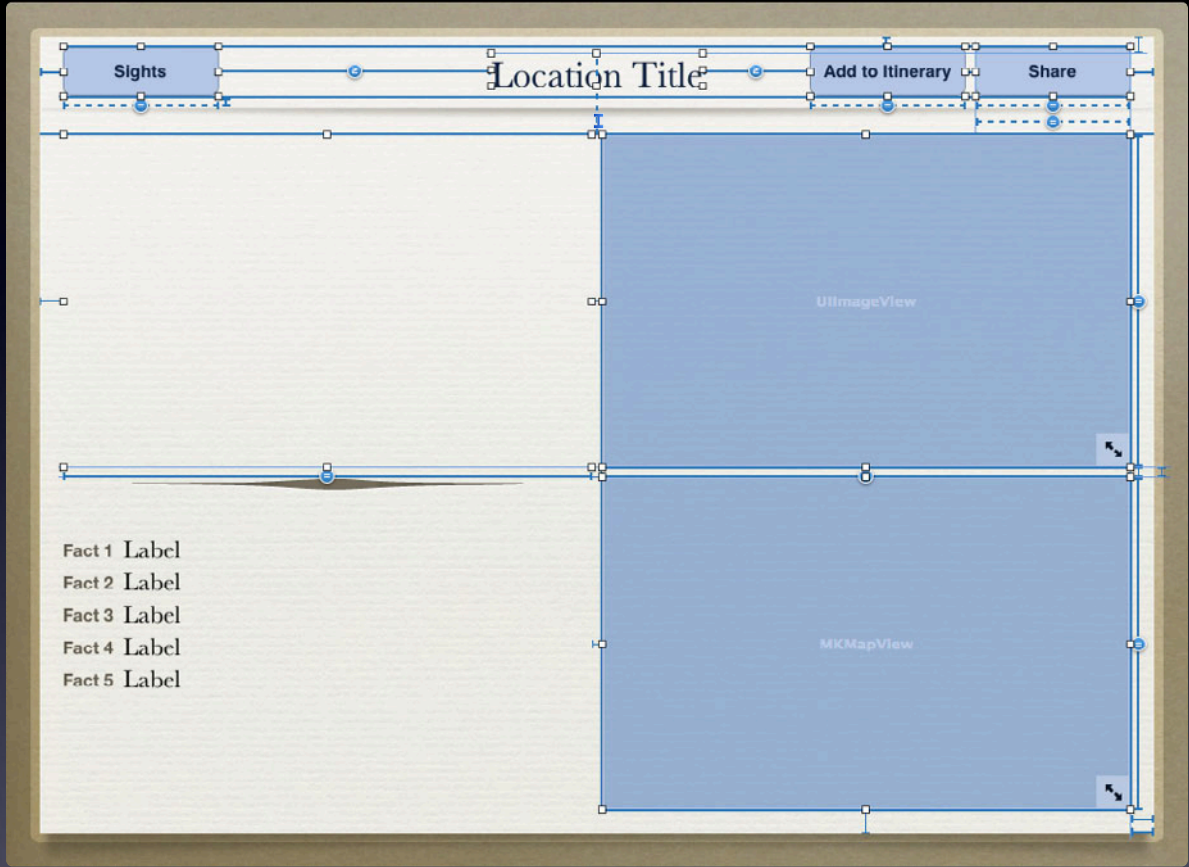
Share

The Golden Gate Bridge is a suspension bridge spanning the Golden Gate, the opening of the San Francisco Bay into the Pacific Ocean. As part of both U.S. Route 101 and California State Route 1, the structure links the city of San Francisco, on the northern tip of the San Francisco Peninsula, to Marin County. It is one of the most internationally recognized symbols of San Francisco, California, and the United States. It has been declared one of the modern Wonders of the World by the American Society of Civil Engineers. The Frommers travel guide considers the Golden Gate Bridge "possibly the most beautiful, certainly the most photographed, bridge in the world".



LENGTH 8,981 ft (2,737.4 m), about 1.7 mi  
WIDTH 90 ft (27.4 m)  
HEIGHT 746 ft (227.4 m)  
LONGEST SPAN 4,200 ft (1,280.2 m)





Localize





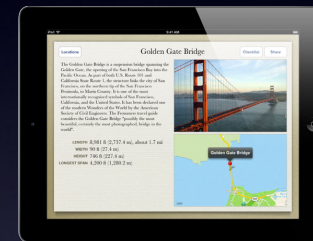
English



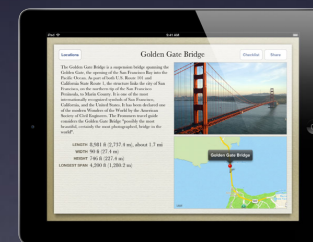
English



French



German



Spanish



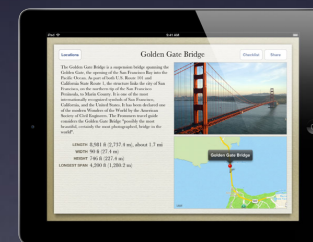
English



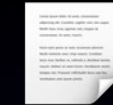
French



German



Spanish



French



English



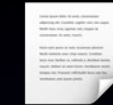
German



Spanish



Base User Interface



French



English



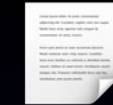
German



Spanish



Base User Interface



French



English



German



Spanish



## Localized Application

# Demonstration

Jon Hess

Senior Engineering Manager, Xcode

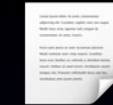




## Localized Application



Base User Interface



French



English



German



Spanish

Design

Optimize

Optim!ze

```
#import "SGTLandmarkDetailViewController.h"
#import "SGTLandmarksViewController.h"
#import "SGTLandmarkTableViewCell.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionSection:
    return [[[SGTLandmarkRepository sharedInstance] landmarks] count];

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];

    NSIndexPath *selectedIndexPath = [[self tableView] indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance]
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell
    [cell setBackgroundColor:[UIColor clearColor]];
}
```

```
#import "SGTLandmarkDetailViewController.h"
#import "SGTLandmarksViewController.h"
#import "SGTLandmarkTableViewCell.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionSection
{
    return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];

    NSIndexPath *selectedLandmarkPath = [[self tableView] indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance] landmarks] objectAtIndex:selectedLandmarkPath.index];
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell
{
    [cell setBackgroundColor:[UIColor clearColor]];
}

```

```
#import "SGTLandmarkDetailViewController.h"
#import "SGTLandmarksViewController.h"
#import "SGTLandmarkTableViewCell.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionInSection
{
    return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];

    NSIndexPath *selectedLandmarkPath = [[self tableView] indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance] landmarks] objectAtIndex:selectedLandmarkPath.index];
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell
{
    [cell setBackgroundColor:[UIColor clearColor]];
}
}
```



```
#import "SGTLandmarkDetailViewController.h"
#import "SGTLandmarksViewController.h"
#import "SGTLandmarkTableViewCell.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionInSection
{
    return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}
```

Issue



Unknown type name "SGTLandmar": did you mean "SGTLandmark"?

sender

Fix-it

Replace "SGTLandmar" with "SGTLandmark"

segue d

```
NSIndexPath *selectedIndexPath = [[self tableView] indexPathForSelectedRow];
SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance]
[detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell
{
    [cell setBackgroundColor:[UIColor clearColor]];
}
```





# LLVM

Chris Lattner

Director, Development Technologies





LLDB



# Apple LLVM Compiler 4







-Weverything

C++

New Loop  
Optimizations

New ARC  
Optimizations

Ivy Bridge Mac Support

Faster ARM Compiles

AVX Vectors

New Warnings

Objective-C Literals



Integrated ARM Assembler

Improved Register Allocation

GC to ARC Migrator

C++





# Xcode 4.2



'auto' variables

Range-based for loops

Rvalue references & move semantics

Variadic templates

Extended SFINAE

Alias templates

Declared type of an expression (decltype)

Default template arguments for function templates

Deleted functions

Extern templates

Inline namespaces

Static assertions

Strongly typed enums

Other: nullptr, override, final, noexcept, ...



Non-static data member initializers  
Raw string literals  
Unicode string literals  
Alignment support - alignas



## Xcode 4.4

Atomics  
Generalized constants - constexpr  
Generalized initializer lists







# Compiler Warnings



```
if (logLength > 0)
{
```

Issue ⚠ Format specifies type 'id' but the argument has type 'GLchar \*' (aka 'char \*')

Fix-it Replace "%@" with "%s"

```
NSLog(@"Shader compile log:\n%s", log);
```

⚠ Format specifies type 'id' but the argument has type 'GLchar \*' (aka 'char \*')

```
free(log);
```





**-Wall**



**-Wall -Wextra -pedantic -Wformat=2**







**-Weverything**





**-Weverything**

**-Wno-extra-tokens**

**-Wno-unused-parameter**

**-Wno-format-y2k**



**-Weverything**

**-Wno-extra-tokens**

**-Wno-unused-parameter**

**-Wno-format-y2k**



# Static Analyzer

# Cross Function Analysis

```
60
61
62
63 void foo(int *P, int Condition) {
64     if (Condition)
65         *P = 0;
66 }
67
68
69 int bar() {
70     int Value;
71
72     foo(&Value, 0);
73
74     return Value;
75 }
76
77
78
```

# Cross Function Analysis

```
60
61
62
63 void foo(int *P, int Condition) {
64     if (Condition)
65         *P = 0;
66 }
67
68
69 int bar() {
70     int Value;
71
72     foo(&Value, 0);
73
74     return Value;
75 }
76
77
78
```

Undefined or garbage value returned to caller

# Cross Function Analysis

```
60
61
62
63 void foo(int *P, int Condition) {
64     if (Condition)
65         *P = 0;
66 }
67
68
69 int bar() {
70     int Value;
71     foo(&Value, 0);
72     return Value;
73 }
74
75
76
77
78
```

Annotations:

- 3. Entered call from 'bar'
- 1. Variable 'Value' declared without an initial value
- 2. Calling 'foo'
- 5. Undefined or garbage value returned to caller

# New Analyzer Checks



GCD

Security

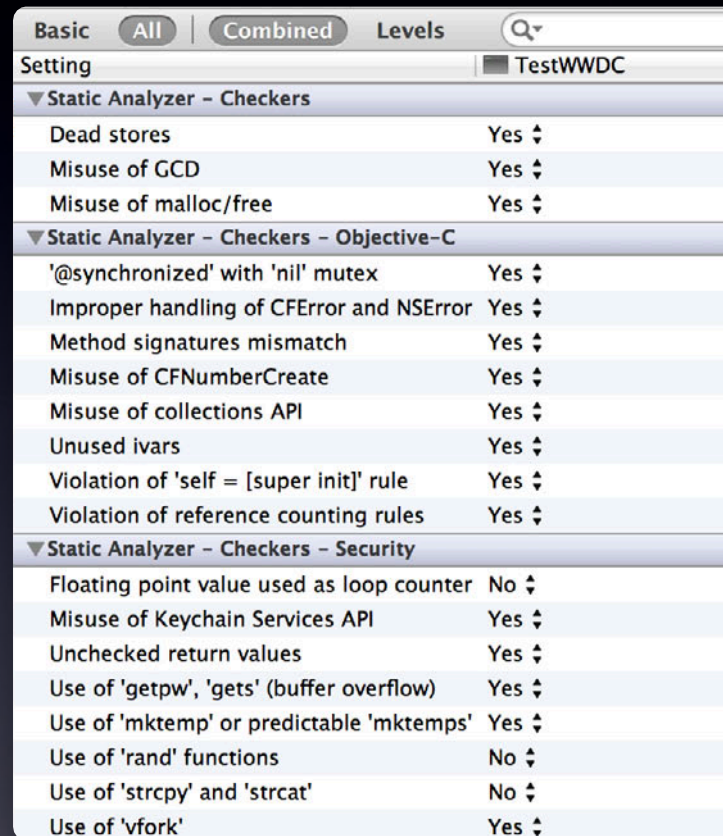
32/64-bit

Unix APIs

malloc/free

Core Foundation

# Select Individual Checkers



The screenshot shows the 'Static Analyzer - Checkers' settings in Xcode. The window has tabs for 'Basic', 'All', 'Combined', and 'Levels', with 'All' selected. A search icon is visible in the top right. Below the tabs, there is a 'Setting' section with a 'TestWWDC' checkbox. The main area is divided into three sections: 'Static Analyzer - Checkers', 'Static Analyzer - Checkers - Objective-C', and 'Static Analyzer - Checkers - Security'. Each section contains a list of checkers with their current status and a toggle arrow.

Checker	Status
<b>Static Analyzer - Checkers</b>	
Dead stores	Yes ⇅
Misuse of GCD	Yes ⇅
Misuse of malloc/free	Yes ⇅
<b>Static Analyzer - Checkers - Objective-C</b>	
'@synchronized' with 'nil' mutex	Yes ⇅
Improper handling of NSError and NSError	Yes ⇅
Method signatures mismatch	Yes ⇅
Misuse of CFNumberCreate	Yes ⇅
Misuse of collections API	Yes ⇅
Unused ivars	Yes ⇅
Violation of 'self = [super init]' rule	Yes ⇅
Violation of reference counting rules	Yes ⇅
<b>Static Analyzer - Checkers - Security</b>	
Floating point value used as loop counter	No ⇅
Misuse of Keychain Services API	Yes ⇅
Unchecked return values	Yes ⇅
Use of 'getpw', 'gets' (buffer overflow)	Yes ⇅
Use of 'mktemp' or predictable 'mktemps'	Yes ⇅
Use of 'rand' functions	No ⇅
Use of 'strcpy' and 'strcat'	No ⇅
Use of 'vfork'	Yes ⇅





The logo for LLDB (Low Level Debugger) is a blue arrow pointing to the right. The letters 'LLDB' are written in a white, sans-serif font inside the arrow. The arrow has a slight gradient and a shadow effect.

LLDB

```
- (BOOL)application:(UIApplication *)application
{
    self.window = [[UIWindow alloc] initWithFrame:
    // Override point for customization after app
    self.viewController = [[ViewController alloc]
```

⏏ | ▶ | ↺ | ⏴ | ⏵ | CoverageTest > Thread

Auto ⌵



All Output ⌵

- ▶ **A** self = (AppDelegate \*) 0x06e1b...
- ▶ **A** launchOptions = (NSDictionary...
- ▶ **A** application = (UIApplication \*) 0...

(lldb)

```
- (BOOL)application:(UIApplication *)application
{
    self.window = [[UIWindow alloc] initWithFrame:
    // Override point for customization after app
    self.viewController = [[ViewController alloc]

```

Navigation icons: Run, Stop, Step Over, Step Into, Step Out, Previous, Next, CoverageTest, Thread

Auto  All Output

- ▶ **A** self = (AppDelegate \*) 0x06e1b...
- ▶ **A** launchOptions = (NSDictionary...
- ▶ **A** application = (UIApplication \*) 0...

```
For more information on any particular command, try
'help <command-name>'.
(lldb) █
```

dmarkParser initWithAttributeDictionary:error:]

All Output ▾

Clear



```
(lldb) po attributes["@coordinates"]
```

```
(id) $1 = 0x09186c60 {37.817582, -122.478719}
```

```
(lldb) p attributes = @{ @"coordinates" : @"{39, -125}", @"distance" : @2500 }
```

```
(NSDictionary *) $2 = 0x09186fb0 2 key/value pairs
```

```
(lldb) po attributes
```

```
(NSDictionary *) $3 = 0x09186fb0 {
```

```
    coordinates = "{39, -125}";
```

```
    distance = 2500;
```

```
}
```

```
(lldb)
```

Sightseer > Thread 1 > 0 -[SGTDataFileLandmarkParser	
Auto	All Out
▶ <b>A</b> self = (SGTDataFileLandmarkParser *) 0x09186d80	(lldb)
▶ <b>A</b> outError = (NSError **) 0xbfffd3d4	(id) \$1
▶ <b>A</b> attributes = (__NSDictionaryM *) 0x09186b30 5 key/value pairs	(lldb)
▶ <b>L</b> coordinatesString = (__NSCFString *) 0x09186c60 @"{37.817582, -122.478719}"	(NSDiction
▶ <b>L</b> localError = (NSError *) 0x00000000	(NSDiction
▶ <b>L</b> point = (CGPoint) (x=37.8176, y=-122.479)	coo
▶ <b>L</b> selBounds = (CGRect) origin=(x=240, y=28) size=(width=47, height=83)	dis
	}
	(lldb)

Print Description of "attributes"  
Copy  
View Value As ▶

Edit Value...  
Edit Summary Format...

Add Expression...  
Delete Expression

**Watch "attributes"**

View Memory of "attributes"  
View Memory of "\*attributes"

✓ Show Summaries  
✓ Show Types

Debug Area Help ▶

Auto ▾

- ▶ **A** self = (SGTDataFileLandmarkParser) 0x09186d80
- ▶ **A** outError = (NSError) 0x00186530
- ▶ **A** attributes = (NSDictionary) {value pairs}
- ▶ **L** coordinatesString = (NSString) "{37.817582, -122.478719}"
- ▶ **L** localError = (NSError) 0x00186530
- ▶ **L** point = (CGPoint) {x=37.8176, y=-122.479}
- ▶ **L** selBounds = (CGRect) {x=240, y=28, size=(width=47, height=83)}

0 -[SGTDataFileLandmarkParser  
All Out  
(lldb)  
(id) \$1  
(lldb)  
(NSDict  
(lldb)  
(NSDict  
coo  
dis  
}  
(lldb)

The LLVM logo is a rounded rectangular button with a metallic, brushed metal texture. The text "LLVM" is centered on the button in a bold, white, sans-serif font.

LLVM

The LLDB logo is a blue arrow-shaped button pointing to the right. The text "LLDB" is centered on the button in a bold, white, sans-serif font.

LLDB





LLVM-GCC



GDB

The LLVM logo is a rounded rectangular button with a metallic, brushed metal texture. The text "LLVM" is centered on the button in a bold, white, sans-serif font.

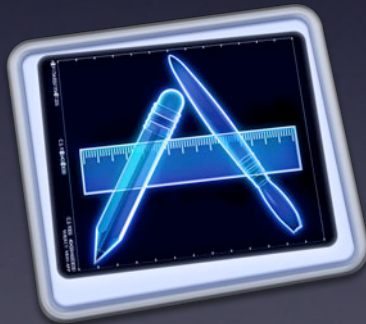
LLVM

The LLDB logo is a blue arrow-shaped button pointing to the right. The text "LLDB" is centered on the button in a bold, white, sans-serif font.

LLDB



LLDB



MobileSafari Target

00:00:12 Run 1 of 1

Threads

- MobileSafari (980) ...thread\_selfid 0x79972
- MobileSafari (980) \_dispatch\_worker\_thread2 0x79968
- MobileSafari (980) \_dispatch\_mgr\_thread\$ARIANT\$mp...
- MobileSafari (980) \_NSThread\_main\_... 0x7998e
- MobileSafari (980) RunWebThread 0x79969
- MobileSafari (980) WTF::wtfThreadEntryPoint 0x799ab
- MobileSafari (980) pthread\_versama\_np 0x799af
- MobileSafari (980)

Trace Highlights

Highlight By: Type of Target

### System Usage

- Preempted
- Runnable
- Supervisor
- Unknown
- Blocked

Category	Percentage
Preempted	94%
Supervisor	0%
Unknown	4%
Blocked	0%

### Thread Context Switches

Thread	Context Switches
_dispatch_mgr_threa...	~1000
WebCore:runLoader...	~1000
Main Thread 0x79963	~1500
__thread_selfid 0x79...	~1800
RunWebThread 0x7...	~2000

### System Call Types

- BSC\_workq\_ops
- BSC\_workq\_ops
- BSC\_kevent64
- MSC\_mach\_ms...
- INTERRUPT

System Call Type	Percentage
BSC_workq_ops	36%
MSC_mach_ms...	25%
INTERRUPT	11%
BSC_kevent64	15%

### Thread Total System Calls

Thread	Total System Calls
_dispatch_worker_thr...	~1690
WebCore:runLoader...	~1690
__thread_selfid 0x79...	~3380
RunWebThread 0x7...	~3380
Main Thread 0x79963	~5070

### Thread Total VM Operations

Thread	Total VM Operations
_dispatch_worker_thr...	~1034
WebCore:runLoader...	~1034
Main Thread 0x79963	~2068
__thread_selfid 0x79...	~3102
RunWebThread 0x7...	~4136

### VM Fault Durations

- Page Cache Hit
- Zero Fill
- Copy On Write
- File Backed Page In

VM Fault Type	Percentage
File Backed Page In	88%
Copy On Write	7%
Zero Fill	0%
Page Cache Hit	0%

### VM Fault Types

- Page Cache Hit
- Copy On Write
- Zero Fill
- File Backed Page In

VM Fault Type	Percentage
File Backed Page In	41%
Copy On Write	22%
Zero Fill	10%
Page Cache Hit	25%

Sightseer.xcodeproj — Sightseer-1.gputrace

Running Sightseer on iPad

No Issues

Sightseer  
Captured OpenGL ES Frame

461 glDrawElements(GL\_TRIANGLE\_STRIP, 88, GL\_UNSIGNED\_SHORT, nullptr)

Bound GL Objects > All > No Selection

- Texture0(2D) #1
- Texture1(2D) #2
- Texture2(2D) #3
- VAO #22
- Vertex Buffer #35
- Index Buffer #36
- Program #30 "metal"
- Renderbuffer #1

2 glPushGroupMarkerEXT(0, "Rendering")  
3 glBindFramebuffer(GL\_FRAMEBUFFER, 4)  
4 glClear(GL\_DEPTH\_BUFFER\_BIT)  
5 glGetIntegerv(GL\_VIEWPORT, {0, 0, ...})  
6 glViewport(0, 0, 2048, 2048)  
7 glEnable(GL\_POI YG.ON\_OFFSFT\_FIL 1)  
8 glPolygonOffset(1.0000000, 1.000...)  
9 glUseProgram(35)  
10 glUniform1f(mtx\_mvp, 0.0000000)  
11 glUniform1f(mtx\_mvp, 0.0000000)  
12 glUniformMatrix4fv(1, 1, 0, {-0.0...})  
13 glUniformMatrix4fv(1, 1, 0, {-0.0...})  
14 glEnable(GL\_DEPTH\_TEST)  
15 glBindVertexArray(391)  
16 glDrawElements(GL\_TRIANGLE\_ST...)  
17 glUseProgram(35)  
18 glUniform1f(mtx\_mvp, 0.0000000)  
19 glUniform1f(mtx\_mvp, 0.0000000)  
20 glUniformMatrix4fv(1, 1, 0, {-0.0...})  
21 glUniformMatrix4fv(1, 1, 0, {-0.0...})  
22 glEnable(GL\_DEPTH\_TEST)  
23 glBindVertexArray(392)  
24 glDrawElements(GL\_TRIANGLE\_ST...)  
25 glUseProgram(35)  
26 glUniform1f(mtx\_mvp, 0.0000000)  
27 glUniform1f(mtx\_mvp, 0.0300000)  
28 glUniformMatrix4fv(1, 1, 0, {-0.0...})  
29 glUniformMatrix4fv(1, 1, 0, {-0.0...})  
30 glEnable(GL\_DEPTH\_TEST)  
31 glBindVertexArray(393)  
32 glDrawElements(GL\_TRIANGLE\_ST...)  
33 glBindVertexArray(394)  
34 glDrawElements(GL\_TRIANGLES)

Color

Auto Buffers View - Zoom to fit Orientation

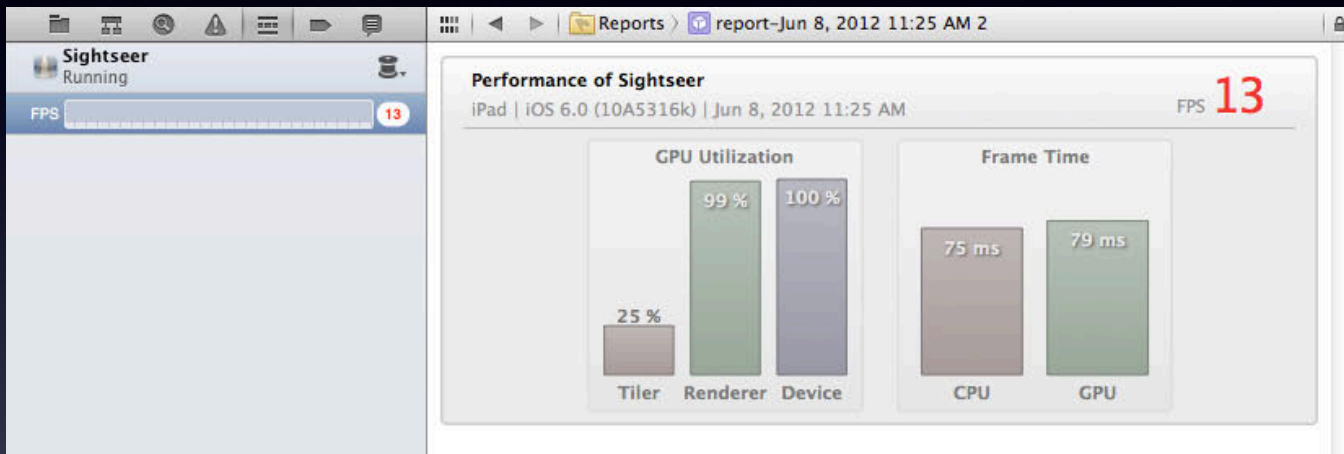
GL Context

- Error Status No GL Error
- Viewport (0, 0, 964, 688) - (0, 1)
- Active Texture Unit Texture Unit 0
- Stencil Off
- Blending Off
- Depth Less - (Write On) - Clear(1.000000)
- Culling Off
- Framebuffer Write - RGBA, Clear (0, 0, 1, 1)
- Polygon Offset Off

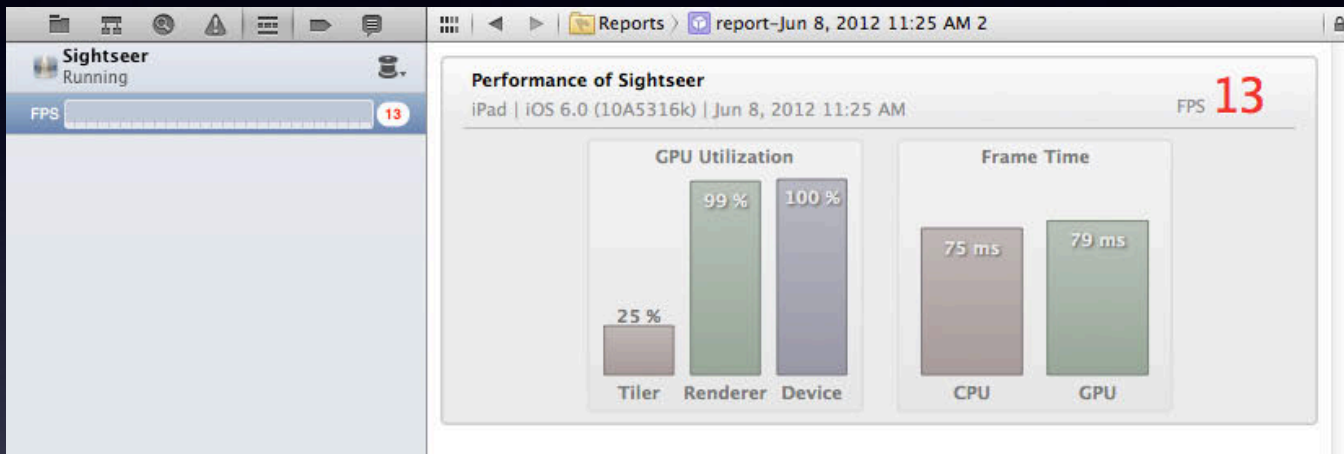
Bound GL Objects

- Program #30 Current Program
- Framebuffer #4 Current Framebuffer
- Renderbuffer #1 Current Renderbuffer
- Buffer #35 (GL\_ARRAY\_BUFFER) Current Array Buffer
- Vertex Array Object #22 Current Vertex Array Object
- Texture Unit #0 2D:#1 Cube Map:#0
- Texture Unit #1 2D:#2 Cube Map:#0
- Texture Unit #2 2D:#3 Cube Map:#0

The image shows a screenshot of a Mac OS X system status menu for the application 'Sightseer'. The menu is centered on a dark blue background. At the top, there is a row of standard system status icons: a folder, a network icon, a Wi-Fi icon, a battery icon, a volume icon, a speaker icon, and a speech bubble icon. Below these icons, the application name 'Sightseer' is displayed in bold, followed by the status 'Running'. To the right of the application name is a small icon of a database cylinder. Below the application name and status is a progress bar labeled 'FPS' on the left and '13' in a red circle on the right. The progress bar is currently empty, indicating 0 FPS. The entire menu has a light gray background and a thin border.







# Demonstration

Ted Kremenek

Senior Engineering Manager, LLVM Compiler

```
60
61
62
63 void foo(int *P, int Condition) {
64     if (Condition)
65         *P = 0;
66 }
67
68
69 int bar() {
70     int Value;
71     foo(&Value, 0);
72     return Value;
73 }
74
75
76
77
78
```

Annotations:

- 3. Entered call from 'bar'
- 1. Variable 'Value' declared without an initial value
- 2. Calling 'foo'
- 5. Undefined or garbage value returned to caller

```
- (BOOL)application:(UIApplication *)application
{
    self.window = [[UIWindow alloc] initWithFrame:
    // Override point for customization after app
    self.viewController = [[ViewController alloc]
    CoverageTest > Thread
    Auto Auto All Output
    ▶ A self = (AppDelegate *) 0x06e1b... (lldb)
    ▶ A launchOptions = (NSDictionary...
    ▶ A application = (UIApplication *) 0...
```

```
For more information on any particular command, try
'help <command-name>'.
(lldb) █
```

Sightseer.xcodeproj — Sightseer-1.gputrace

Running Sightseer on iPad

No Issues

Run Stop Scheme Breakpoints Editor View Organizer

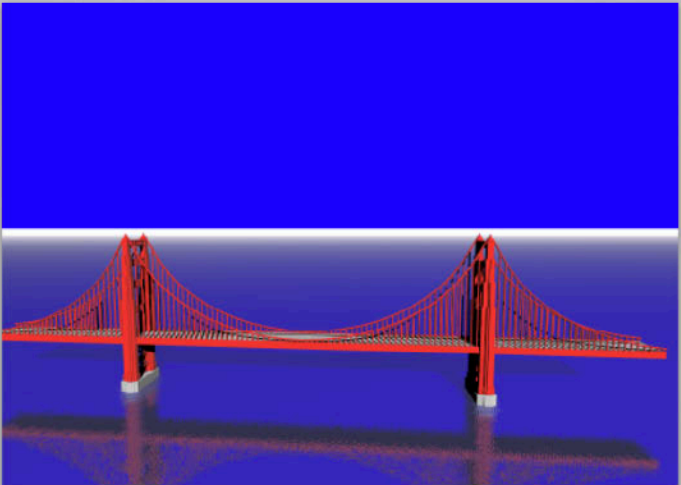
461 glDrawElements(GL\_TRIANGLE\_STRIP, 88, GL\_UNSIGNED\_SHORT, nullptr)

Bound GL Objects > All > No Selection

Sightseer  
Captured OpenGL ES Frame

rrc

- Sightseer 3D | com.apple.main-thread
  - 0 glBindFramebuffer(GL\_FRAMEBUFFER, 4)
  - 1 glViewport(0, 0, 964, 688)
  - Rendering
    - 2 glPushGroupMarkerEXT(0, "Rendering")
    - 3 glBindFramebuffer(GL\_FRAMEBUFFER, 4)
    - 4 glClear(GL\_DEPTH\_BUFFER\_BIT)
    - 5 glGetIntegerv(GL\_VIEWPORT, {0, 0, 964, 688})
    - 6 glViewport(0, 0, 2048, 2048)
    - 7 glEnable(GL\_POINT\_SPRITE)
    - 8 glPolygonOffset(1.0000000, 1.0000000)
    - 9 glUseProgram(35)
    - 10 glUniform1f(mtx\_mvp, 0.0000000)
    - 11 glUniform1f(mtx\_mvp, 0.0000000)
    - 12 glUniformMatrix4fv(1, 1, 0, {-0.0000000, 0.0000000, 0.0000000, 0.0000000})
    - 13 glUniformMatrix4fv(1, 1, 0, {-0.0000000, 0.0000000, 0.0000000, 0.0000000})
    - 14 glEnable(GL\_DEPTH\_TEST)
    - 15 glBindVertexArray(391)
    - 16 glDrawElements(GL\_TRIANGLE\_STRIP, 88, GL\_UNSIGNED\_SHORT, nullptr)
    - 17 glUseProgram(35)
    - 18 glUniform1f(mtx\_mvp, 0.0000000)
    - 19 glUniform1f(mtx\_mvp, 0.0000000)
    - 20 glUniformMatrix4fv(1, 1, 0, {-0.0000000, 0.0000000, 0.0000000, 0.0000000})
    - 21 glUniformMatrix4fv(1, 1, 0, {-0.0000000, 0.0000000, 0.0000000, 0.0000000})
    - 22 glEnable(GL\_DEPTH\_TEST)
    - 23 glBindVertexArray(392)
    - 24 glDrawElements(GL\_TRIANGLE\_STRIP, 88, GL\_UNSIGNED\_SHORT, nullptr)
    - 25 glUseProgram(35)
    - 26 glUniform1f(mtx\_mvp, 0.0000000)
    - 27 glUniform1f(mtx\_mvp, 0.0300000)
    - 28 glUniformMatrix4fv(1, 1, 0, {-0.0000000, 0.0000000, 0.0000000, 0.0000000})
    - 29 glUniformMatrix4fv(1, 1, 0, {-0.0000000, 0.0000000, 0.0000000, 0.0000000})
    - 30 glEnable(GL\_DEPTH\_TEST)
    - 31 glBindVertexArray(393)
    - 32 glDrawElements(GL\_TRIANGLE\_STRIP, 88, GL\_UNSIGNED\_SHORT, nullptr)
    - 33 glBindVertexArray(394)
    - 34 glDrawElements(GL\_TRIANGLES, 88, GL\_UNSIGNED\_SHORT, nullptr)



Color

Auto Buffers View - Zoom to fit Orientation

GL Context

- Error Status No GL Error
- Viewport (0, 0, 964, 688) - (0, 1)
- Active Texture Unit Texture Unit 0
- Stencil Off
- Blending Off
- Depth Less - (Write On) - Clear(1.0000000)
- Culling Off
- Framebuffer Write - RGBA, Clear (0, 0, 1, 1)
- Polygon Offset Off

Bound GL Objects

- Program #30 Current Program
- Framebuffer #4 Current Framebuffer
- Renderbuffer #1 Current Renderbuffer
- Buffer #35 (GL\_ARRAY\_BUFFER) Current Array Buffer
- Vertex Array Object #22 Current Vertex Array Object
- Texture Unit #0 2D:#1 Cube Map:#0
- Texture Unit #1 2D:#2 Cube Map:#0
- Texture Unit #2 2D:#3 Cube Map:#0

Texture0(2D) #1 Texture1(2D) #2 Texture2(2D) #3 VAO #22 Vertex Buffer #35

Index Buffer #36 Program #30 "metal" Renderbuffer #1

Sightseer.xcodeproj  
Running Sightseer on iPad

Run Stop Scheme Breakpoints No Issues Editor View Organizer

Reports > report-Jun 8, 2012 11:27 AM 2

Aut... > Program #31 "Water Effect" > Fragment Shader

Sightseer  
Captured OpenGL ES Frame

13

Sightseer 3D | com.apple.main-thread

- 0 glBindFramebuffer(GL\_FRAMEBUFFER, 4)
- 1 glViewport(0, 0, 964, 688)
- Rendering
  - 2 glPushGroupMarkerEXT(0, "Rendering")
  - 3 glBindFramebuffer(GL\_FRAMEBUFFER, 4)
  - 4 glClear(GL\_DEPTH\_BUFFER\_BIT)
  - 5 glGetIntegerv(GL\_VIEWPORT, {0, 0, ...})
  - 6 glViewport(0, 0, 2048, 2048)
  - 7 glFenable(GL\_POLYGON\_OFFSET\_FILL)
  - 8 glPolygonOffset(1.0000000, 1.0000000)
  - 9 glUseProgram(35)
  - 10 glUniform1f(mtx\_mv, 0.0000000)
  - 11 glUniform1f(mtx\_mv, 0.0000000)
  - 12 glUniformMatrix4fv(1, 1, 0, {-0.0...})
  - 13 glUniformMatrix4fv(1, 1, 0, {-0.0...})
  - 14 glEnable(GL\_DEPTH\_TEST)
  - 15 glBindVertexArray(391)
  - 16 glDrawElements(GL\_TRIANGLES, ...)
  - 17 glUseProgram(35)
  - 18 glUniform1f(mtx\_mv, 0.0000000)
  - 19 glUniform1f(mtx\_mv, 0.0000000)
  - 20 glUniformMatrix4fv(1, 1, 0, {-0.0...})
  - 21 glUniformMatrix4fv(1, 1, 0, {-0.0...})
  - 22 glEnable(GL\_DEPTH\_TEST)
  - 23 glBindVertexArray(392)
  - 24 glDrawElements(GL\_TRIANGLES, ...)
  - 25 glUseProgram(35)
  - 26 glUniform1f(mtx\_mv, 0.0000000)
  - 27 glUniform1f(mtx\_mv, 0.0300000)
  - 28 glUniformMatrix4fv(1, 1, 0, {-0.0...})
  - 29 glUniformMatrix4fv(1, 1, 0, {-0.0...})
  - 30 glFenable(GL\_DEPTH\_TEST)
  - 31 glBindVertexArray(393)
  - 32 glDrawElements(GL\_TRIANGLES, ...)
  - 33 glBindVertexArray(394)
  - 34 glDrawElements(GL\_TRIANGLES, ...)

Performance of Sightseer  
iPad | iOS 6.0 (10A5316k) | Jun 8, 2012 11:32 AM FPS 13 Target 60

Frame rate of your app is limited by the graphics pipeline.

GPU Utilization

Tiler	25 %	Renderer	98 %	Device	99 %
-------	------	----------	------	--------	------

Frame Time

CPU	64 ms	GPU	75 ms
-----	-------	-----	-------

Fragment Shading  
Performance is limited by fragment shaders

- The program with the most costly fragment shader is:
  - Program #31 "Water Effect"
- The fragment shader in Program #3 "metal" performed dependent texture reads, which are slower than non-dependent texture reads.
  - Program #3 "metal"
    - 129 glDrawElements(GL\_TRIANGLES, 10, GL\_UNSIGNED\_SHORT, nullptr)
    - 119 glDrawElements(GL\_TRIANGLES, 10, GL\_UNSIGNED\_SHORT, nullptr)

Analyze Performance

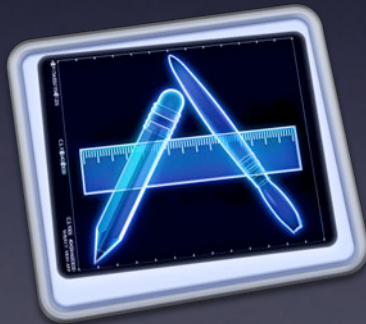
GL Context: No Selection

Bound GL Objects

```

29 varying highp vec4 shadowSpacePos;
30 uniform highp float wavePhase;
31 uniform sampler2D shadow_map;
32 uniform highp float shadow_depth_bias;
33 uniform highp vec3 sun_dir;
34 uniform sampler2D reflection_map;
35 uniform sampler2D reflection_depth_map;
36 varying highp vec4 worldPos;
37 uniform highp vec3 camPos, camDir, camUp;
38
39 highp vec4 shadow()
40 {
41     highp float fshadow = shadow2DProjEXT(shadow_map, shadowSpacePos);
42     return vec4(fshadow) * 0.3 + 0.7;
43 }
44
45 void main()
46 {
47     highp vec3 side = cross(camDir, camUp);
48
49     highp vec3 v = worldPos.xyz - camPos;
50     highp float vz = dot(v, camDir);
51     v = v / vz;
52
53     highp vec2 tc = vec2(dot(v, side), dot(v, camUp)) / TAN_60;
54     tc = tc * 0.5 + 0.5;
55
56     highp vec2 p = modelPos;
57     highp vec2 tcOffset = vec2(0.0, 0.0);
58
59     if (waveCount > 0)
60     {
61         highp float theta = length(vec2(p.x - 0.1, p.y)) * 1000.0;
62         tcOffset += vec2(sin(theta), cos(theta));
63     }
64
65     if (waveCount > 1)
66     {

```







Optim!ze

Deploy

Deploy y



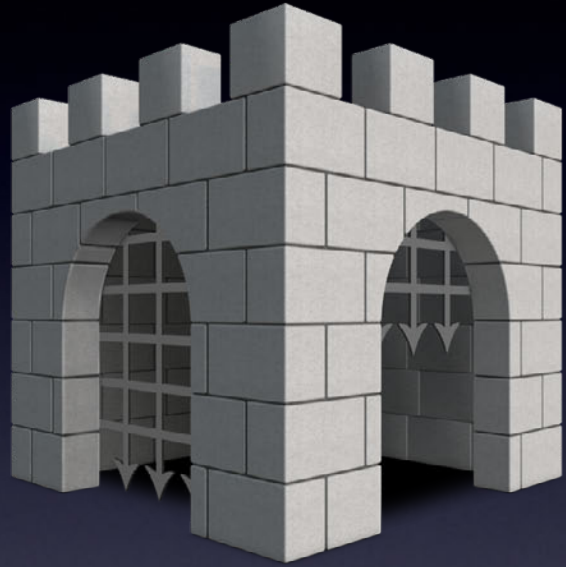


App Store



Developer ID







App Store



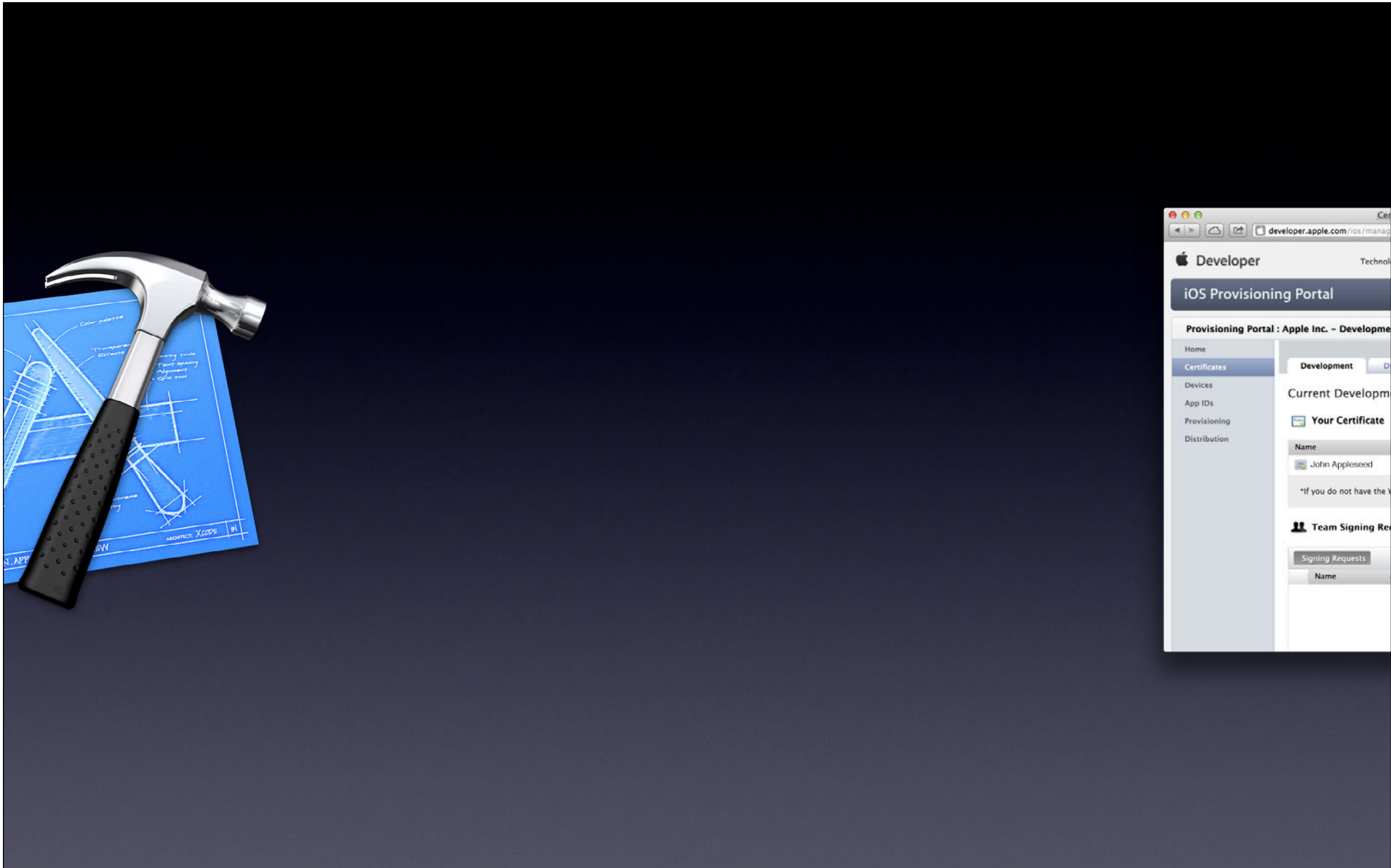
Developer ID





App Store





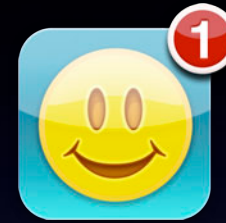




Game Center



iCloud



Push Notifications



In App Purchase



Passbook



Transit App







## ▼ Entitlements

Enable App Sandbox

### Network

Allow Incoming Network Connections

Allow Outgoing Network Connections

### Hardware

Allow Camera Access

Allow Microphone Access

Allow USB Access

Allow Printing

### Apps

Allow Address Book Data Access

Allow Location Services Access

Allow Calendar Data Access





Xcode File Edit View Navigate Editor Product Window

Sightseer.xcodeproj — .m

Analyze Succeeded

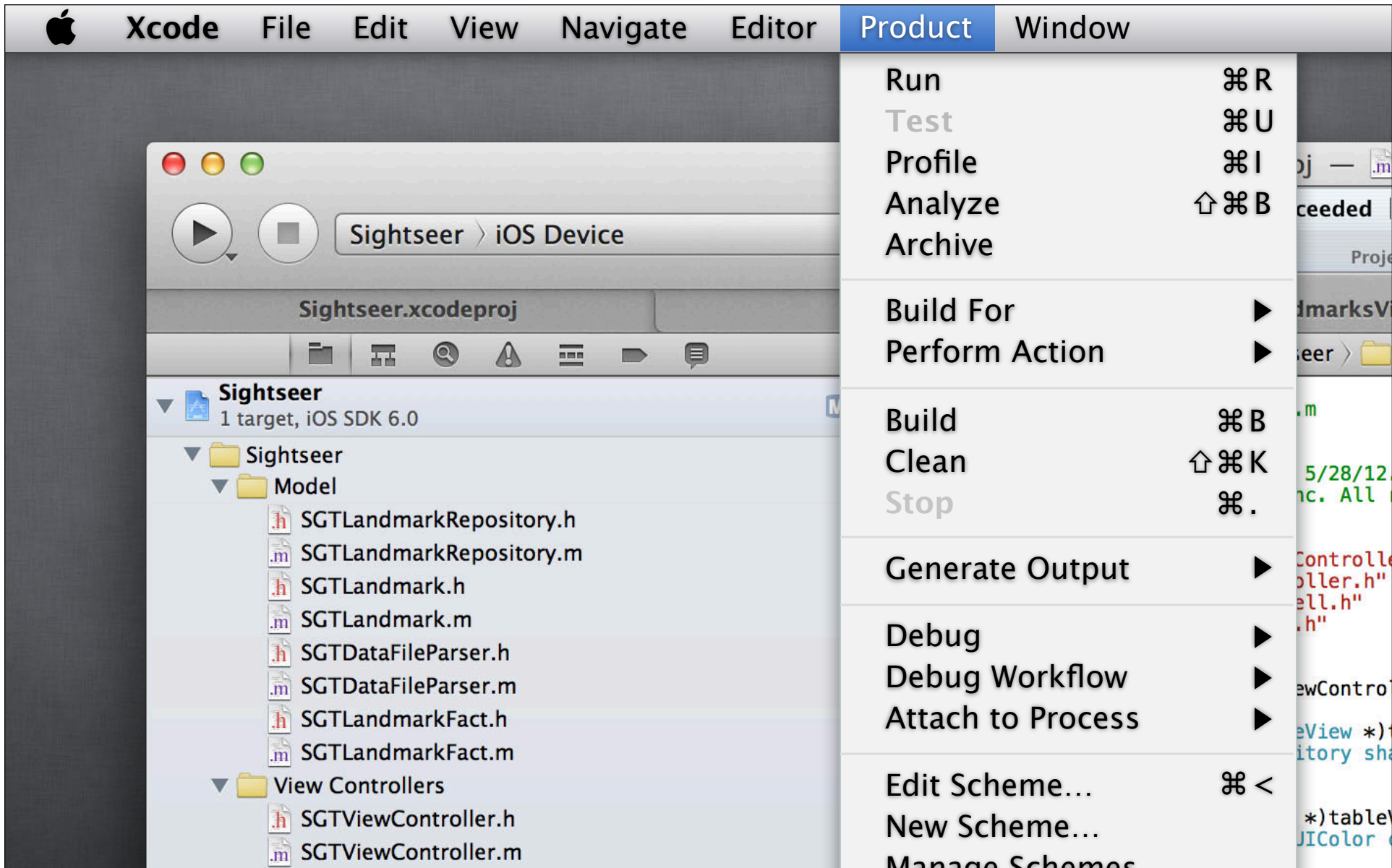
Sightseer > iOS Device

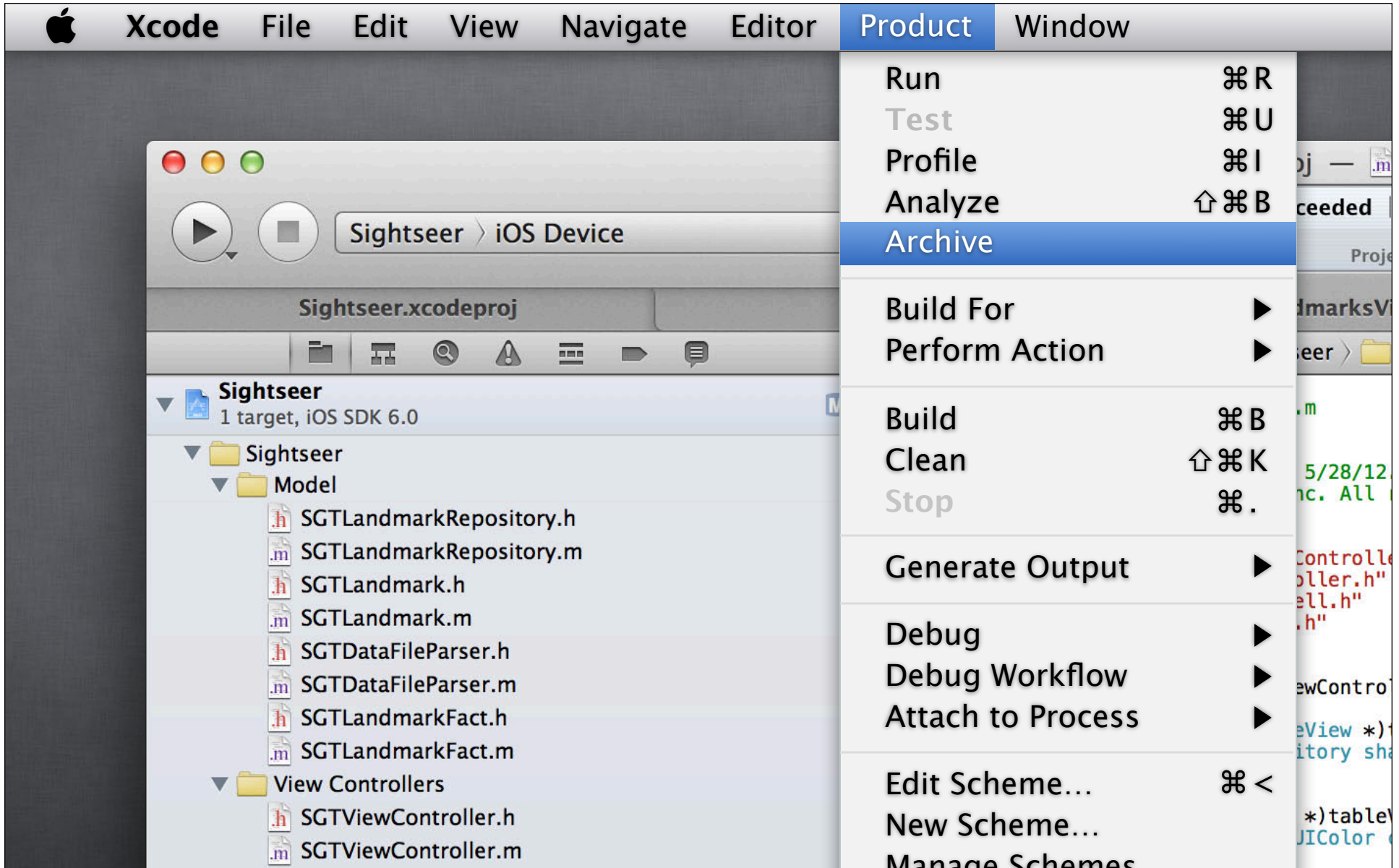
Sightseer.xcodeproj Debugger SGTLandmarksV

Sightseer  
1 target, iOS SDK 6.0

- Sightseer
  - Model
    - SGTLandmarkRepository.h
    - SGTLandmarkRepository.m
    - SGTLandmark.h
    - SGTLandmark.m
    - SGTDataFileParser.h
    - SGTDataFileParser.m
    - SGTLandmarkFact.h
    - SGTLandmarkFact.m
  - View Controllers
    - SGTViewController.h
    - SGTViewController.m

```
1 //
2 // SGTLocationsViewController.m
3 // Sightseer
4 //
5 // Created by Kevin Cathey on 5/28/12
6 // Copyright (c) 2012 Apple Inc. All rights reserved.
7 //
8
9 #import "SGTLandmarkDetailViewController.h"
10 #import "SGTLandmarksViewController.h"
11 #import "SGTLandmarkTableViewCell.h"
12 #import "SGTLandmarkRepository.h"
13 #import "SGTLandmark.h"
14
15 @implementation SGTLandmarksViewController
16
17 - (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:NSInteger {
18     return [[SGTLandmarkRepository sharedInstance] count];
19 }
20
21 - (void)tableView:(UITableView *)tableView cellForRowAtIndexPath:UITableViewCellIdentifier {
22     [cell setBackgroundColor:[UIColor whiteColor]];
23 }
```







## Sightseer

Archive Type: iOS App Archive

Creation Date: June 12, 2012 9:45AM

Version: 1.4

Identifier: com.apple.sightseer

Estimated App Store Size: 15MB

Validate...

Distribute...

🔍 Name

Name	Creation Date	Comment	Status
SightSeer	June 8, 2012 2:07PM		Submitted
Sightseer	June 10, 2012 6:22PM		Validated
Sightseer	June 11, 2012 9:45AM		



## Sightseer

Archive Type: iOS App Archive

Creation Date: June 12, 2012 9:45AM

Version: 1.4

Identifier: com.apple.sightseer

Estimated App Store Size: 15MB

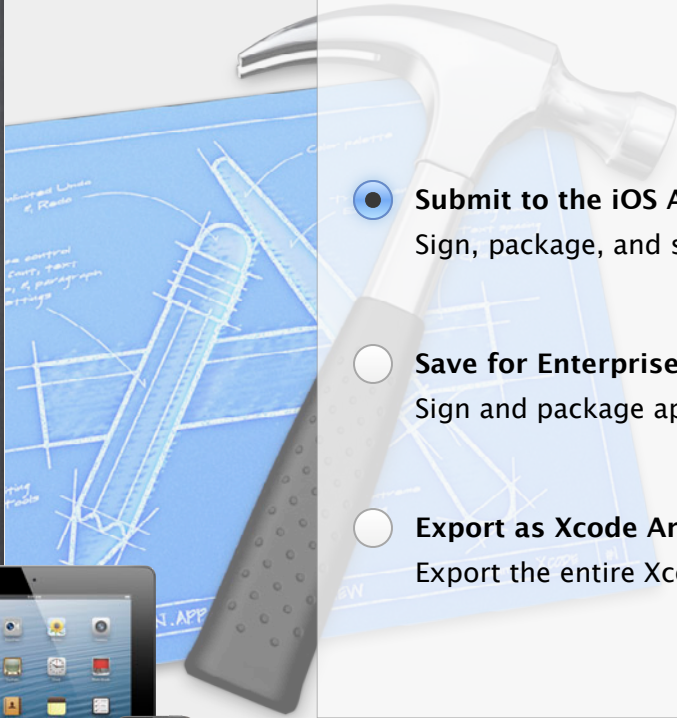
Validate...

Distribute...

🔍 Name

Name	Creation Date ▼	Comment	Status
SightSeer	June 8, 2012 2:07PM		Submitted
Sightseer	June 10, 2012 6:22PM		Validated
Sightseer	June 11, 2012 9:45AM		

Select the method of distribution:

- 
- Submit to the iOS App Store**  
Sign, package, and submit application to the iOS App Store.
  - Save for Enterprise or Ad-Hoc Deployment**  
Sign and package application for distribution outside of the iOS App Store.
  - Export as Xcode Archive**  
Export the entire Xcode archive of this application

Cancel

Previous

Next



Select the method of distribution:

- Submit to the Mac App Store**  
Sign, package, and submit application to the Mac App Store.
- Export Developer ID-signed Application**  
Save a copy of the application signed with your Developer ID.
- Export as** Xcode Archive

Cancel

Previous

Next





Beta Testing



Mac App Store



iOS App Store



Developer ID



Ad-Hoc Deployment



Enterprise Deployment





Beta Testing



Mac App Store



iOS App Store



Developer ID



Ad-Hoc Deployment



Enterprise Deployment

**Now**





# Available Now



# Available Now



 WWDC2012

