

Developer Tools Kickoff

Matthew Firlik
Director, Development Technologies





Xcode 4.0

March 2011







4.1

Auto Layout for OS X
Custom Behaviors
Preprocess/Assembly Assistant



4.2

iOS Storyboards

Automatic Reference Counting

OpenGL ES Debugger



4.3

Developer ID

Self-Contained Packaging

Separate Tools Packages











Graphics Tools



Audio Tools



Hardware Tools

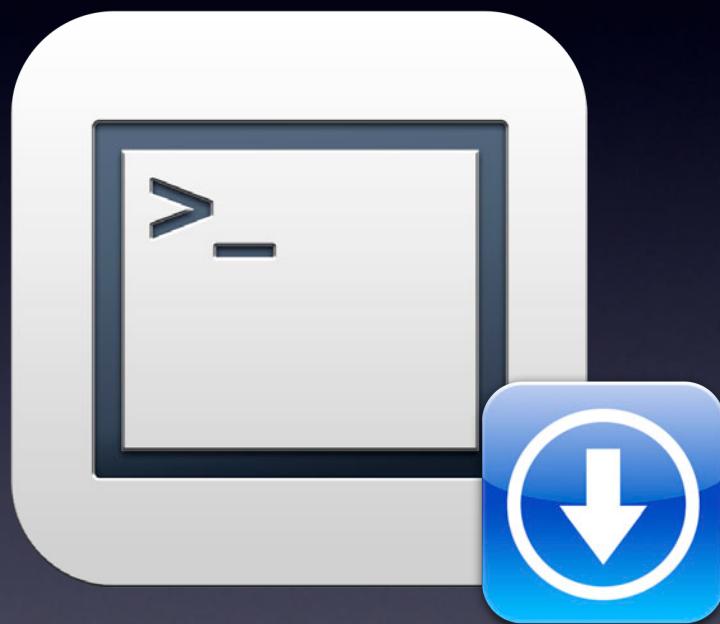


Accessibility Tools



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.

...More

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

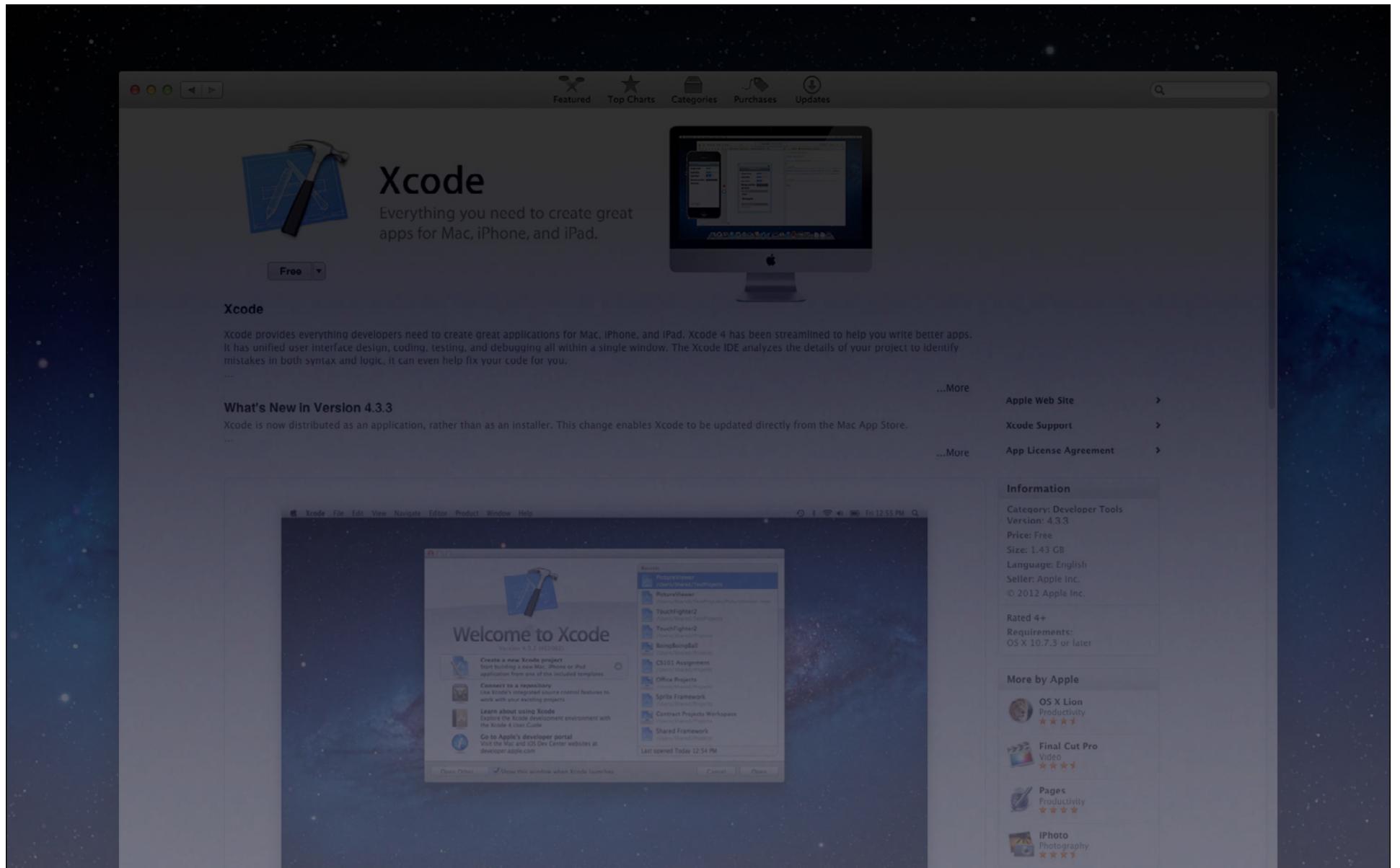
 **OS X Lion**
Productivity ★★★★☆

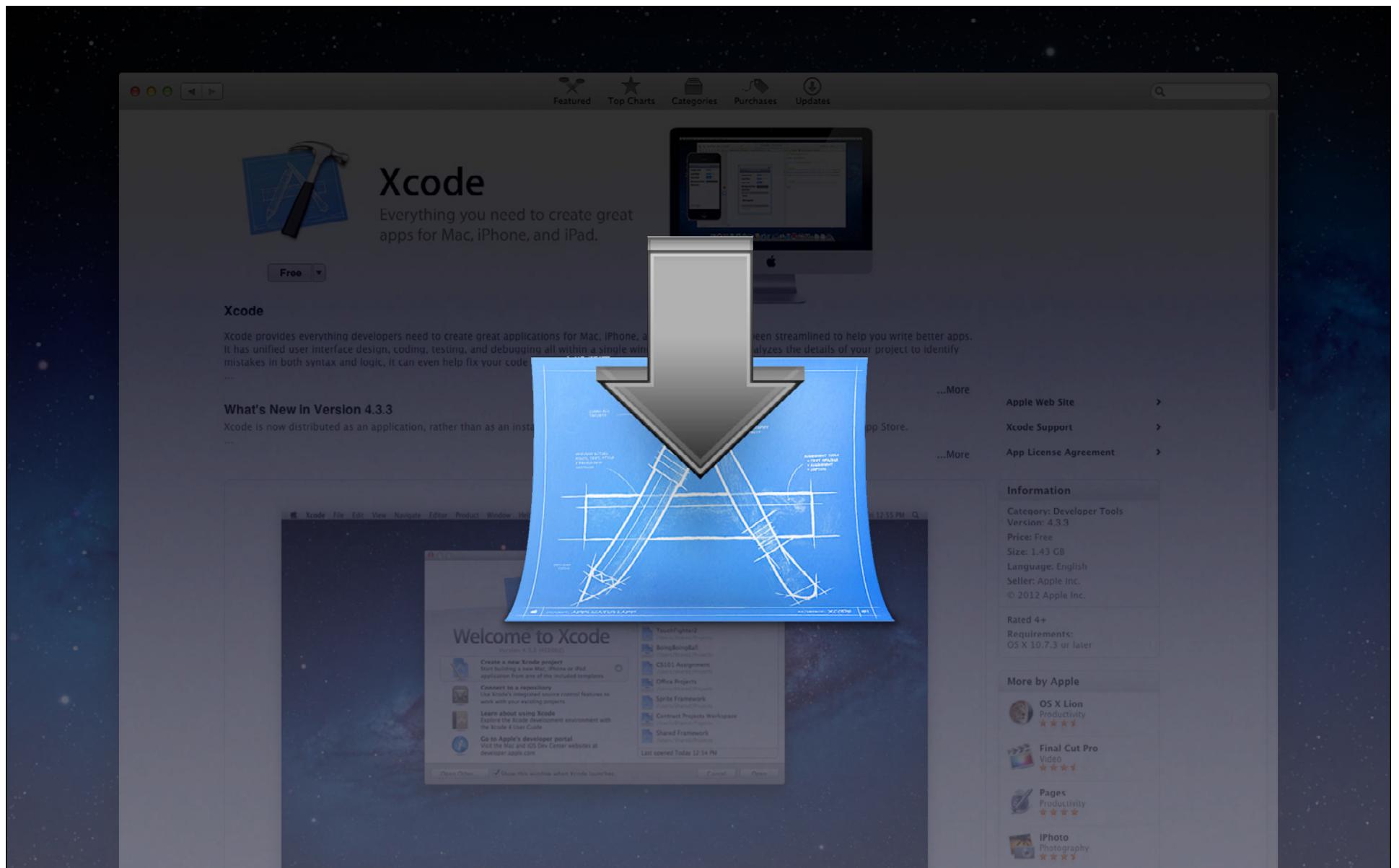
 **Final Cut Pro**
Video ★★★★☆

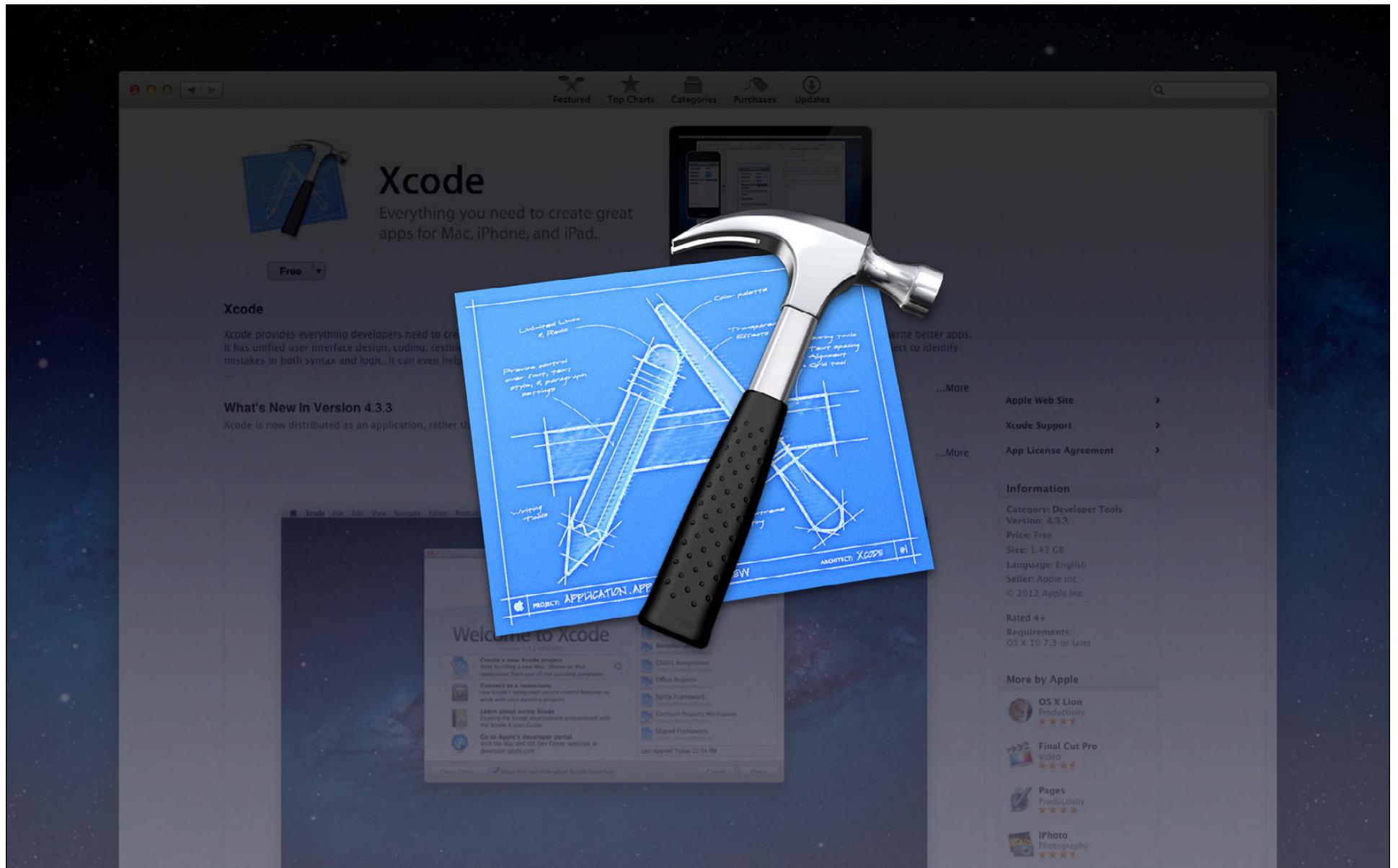
 **Pages**
Productivity ★★★★☆

 **iPhoto**
Photography ★★★★☆

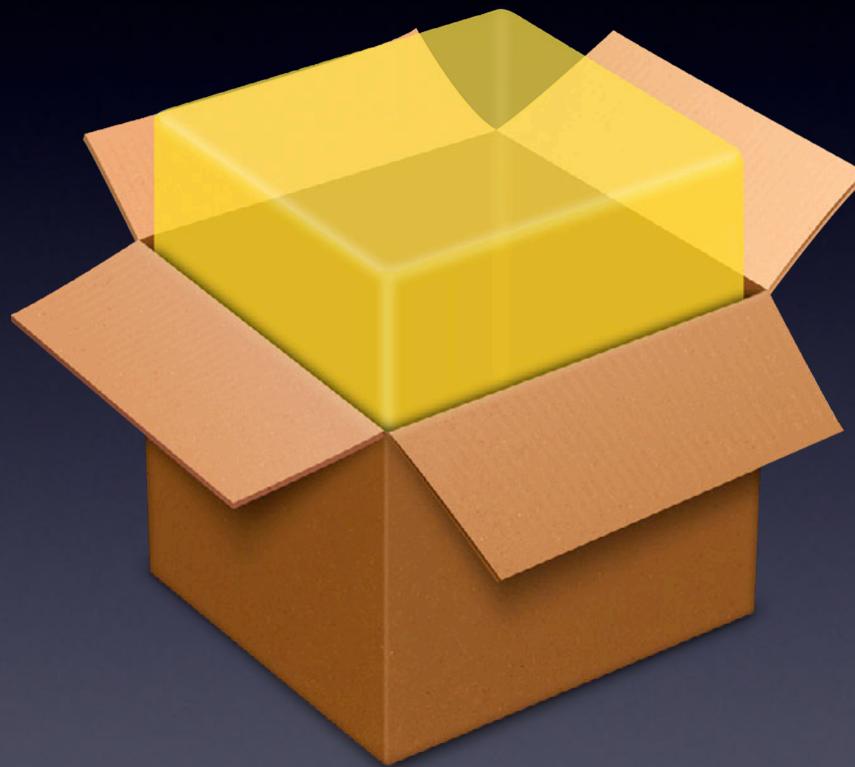






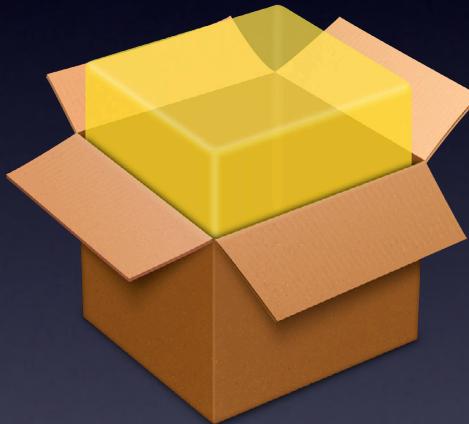


Xcode 4.1
4GB



Xcode 4.2

1.5GB

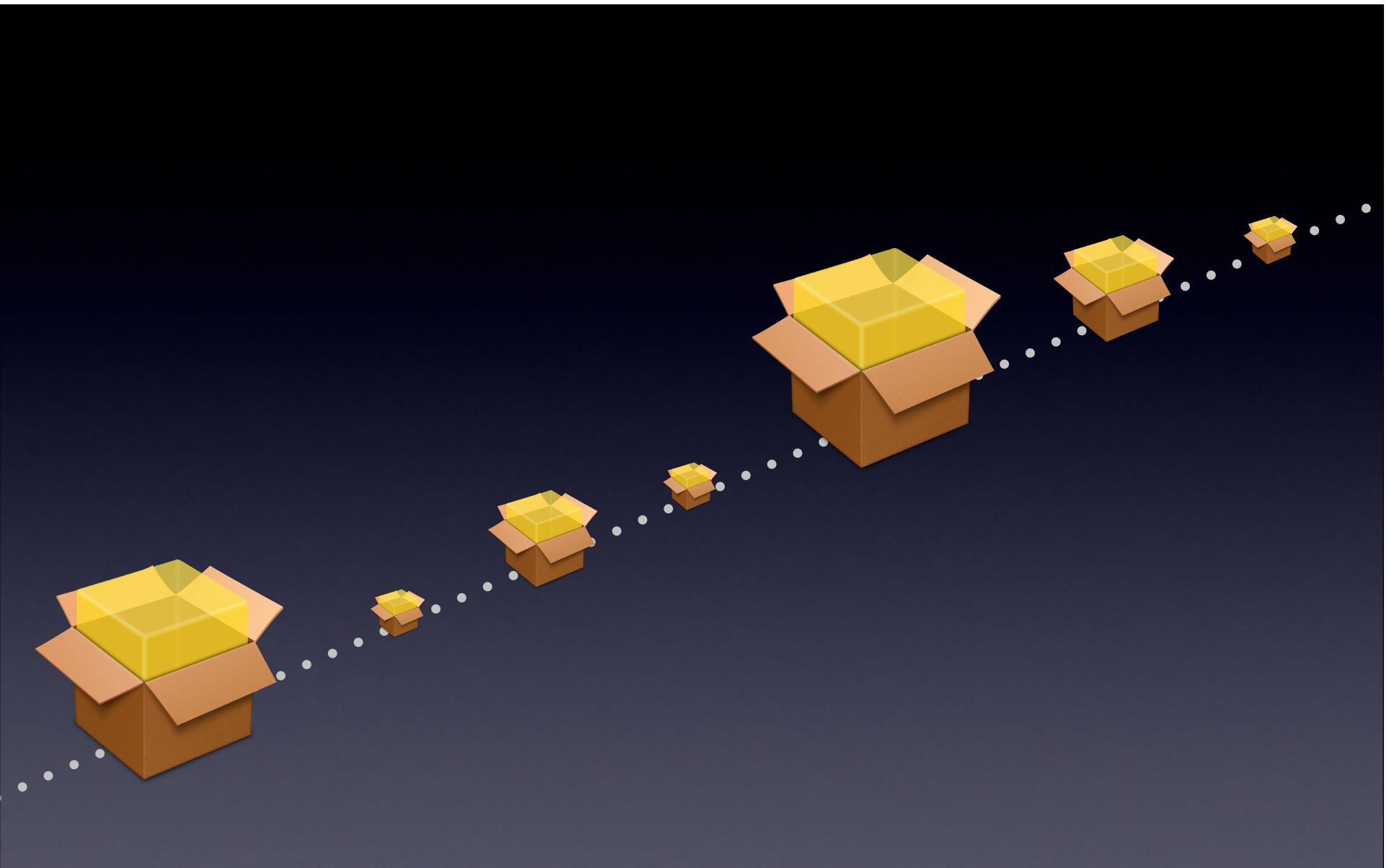


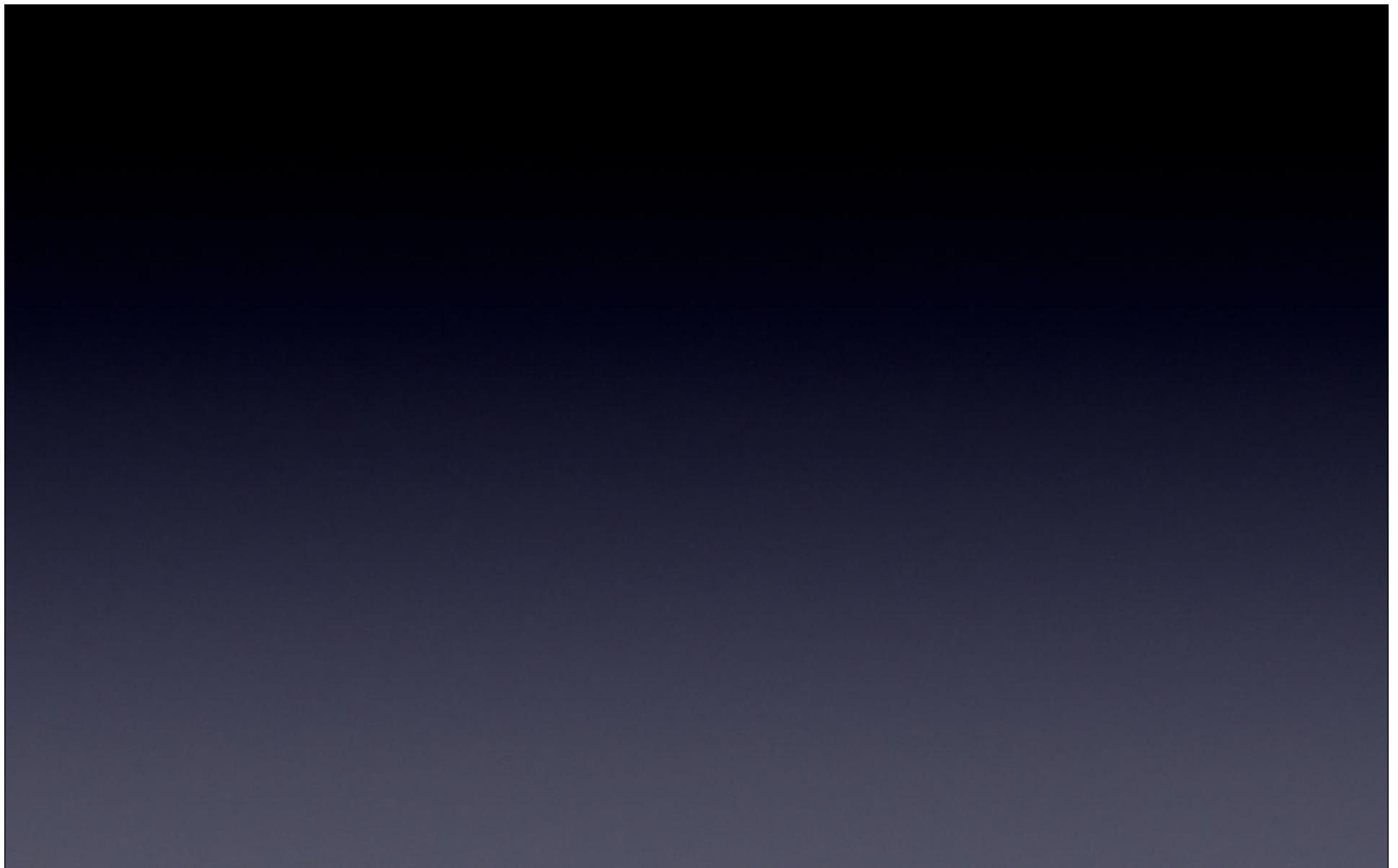
Xcode 4.3.3

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

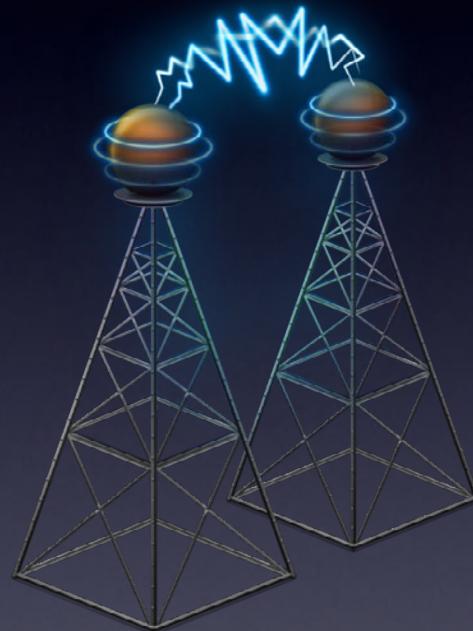
Demonstration

Mike Ferris
Senior Engineering Manager, Xcode

@ literals

`@property`
`@synthesize by default`

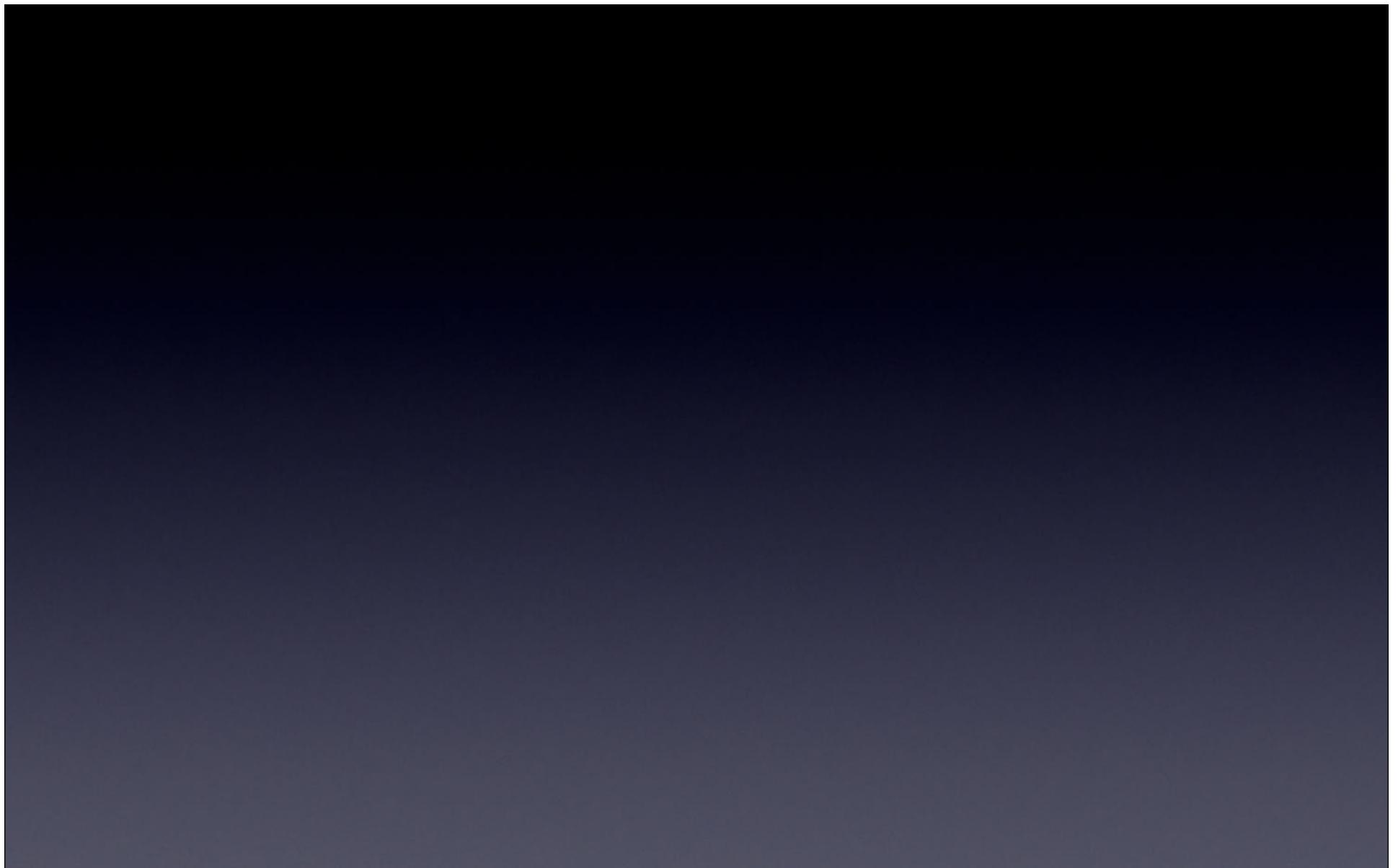
Migration Support for Garbage Collection



L@nguage

Edit

EdIt



```
#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] objectAtIndex:selectedLandmarkPath.row];
    [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.landmark = [SGTLandmarkRepository sharedRepository].landmarks[selectedLandmarkPath];
}

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

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.landmark = [SGTLandmarkRepository sharedRepository].landmarks[selectedLandmarkPath];
}

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

C NSIndexPath

C NSIndexSet

The NSIndexSet 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 *selectedLandmarkPath = [[self tableView] indexPathForSelectedRow];
    detailViewController.landmark = [SGTLandmarkRepository sharedRepository].landmarks[selectedLandmarkPath];
}

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

C NSIndexPath

C NSIndexSet

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

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

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

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
    SGTLandmark *landmark = [[[SGTLandmarkRepository sharedInstance] landmarks] objectAtIndex:indexPath.row];
    SGTLandmarkCell *cell = [tableView dequeueReusableCellWithIdentifier:@"SGTLandmarkCell" forIndexPath:indexPath];
    cell.landmark = landmark;
    return cell;
}
```

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] landmarks] objectAtIndex:selectedLandmarkPath.row];
    [detailViewController setLandmark:selectedLandmark];
}

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

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

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
    SGTLandmark *landmark = [[[SGTLandmarkRepository sharedInstance] landmarks] objectAtIndex:indexPath.row];
    SGTLandmarkCell *cell = [tableView dequeueReusableCellWithIdentifier:@"SGTLandmarkCell" forIndexPath:indexPath];
    cell.landmark = landmark;
    return cell;
}
```

Improved Quick Help

alt

option

?

- `(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.row];`
`[detailViewController setLandmark:selectedLandmark];`
}
- `(void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell forRowAtIndexPath:(NSIndexPath *)indexPath`
[cell setBackgroundColor:[UIColor clearColor]];
}
- `(NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section`
return [[[SGTLandmarkRepository sharedInstance] landmarks] count];
}
- `(UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath`
[cell.textLabel setText:[[[SGTLandmarkRepository sharedInstance] landmarks] objectAtIndex:indexPath.row].name];
[cell.detailTextLabel setText:[[[SGTLandmarkRepository sharedInstance] landmarks] objectAtIndex:indexPath.row].description];
[cell.imageView setImage:[SGTImageCache sharedImageCache].imageForLandmark:[[[SGTLandmarkRepository sharedInstance] landmarks] objectAtIndex:indexPath.row]];
return cell;

Improved Quick Help

alt

option

- `(void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender`
[super prepareForSegue:segue sender:sender];
SGTHandyStoryboardSegueController <detailViewController> = [segue destinationViewController];
NSLog(@"%@", detailViewController);
SGTHandyStoryboardSegueController <detailViewController> = [segue destinationViewController];
[detailViewController release];
[de...]
}
- 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)

- ([NSInterface](#)) [referenceForStoryboardSegueWithIdentifier:](#)
 Declared [UIViewController.h](#)
 Reference [UIViewController Reference](#)
- ([UITableViewCell](#) *) [tableView:\(UITableView *\)tableView cellForRowAtIndexPath:\(NSIndexPath *\)indexPath](#)

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] landmarks] objectAtIndex:selectedLandmarkPath.row];
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell forRowAtIndexPath:(NSIndexPath *)indexPath
{
    [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"];
    if (!cell)
    {
        cell = [[[NSBundle mainBundle] loadNibNamed:@"SGTLandmarkTableViewCell" owner:self options:nil] objectAtIndex:0];
    }
    cell.landmark = [[[SGTLandmarkRepository sharedInstance] landmarks] objectAtIndex:indexPath.row];
    return cell;
}
```

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] landmarks] objectAtIndex:selectedLandmarkPath.row];
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell forRowAtIndexPath:(NSIndexPath *)indexPath
{
    [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"];
    if (!cell)
    {
        cell = [[SGTLandmarkTableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:@"SGTLandmarkTableViewCell"];
    }
    cell.landmark = [[[SGTLandmarkRepository sharedInstance] landmarks] objectAtIndex:indexPath.row];
    return cell;
}
```

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] landmarks] objectAtIndex:selectedLandmarkPath.row];
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell forRowAtIndexPath:(NSIndexPath *)indexPath
{
    [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
```

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] landmarks] objectAtIndex:selectedLandmarkPath.row];
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell forRowAtIndexPath:(NSIndexPath *)indexPath
{
    [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
```

Trailing Whitespace Removal

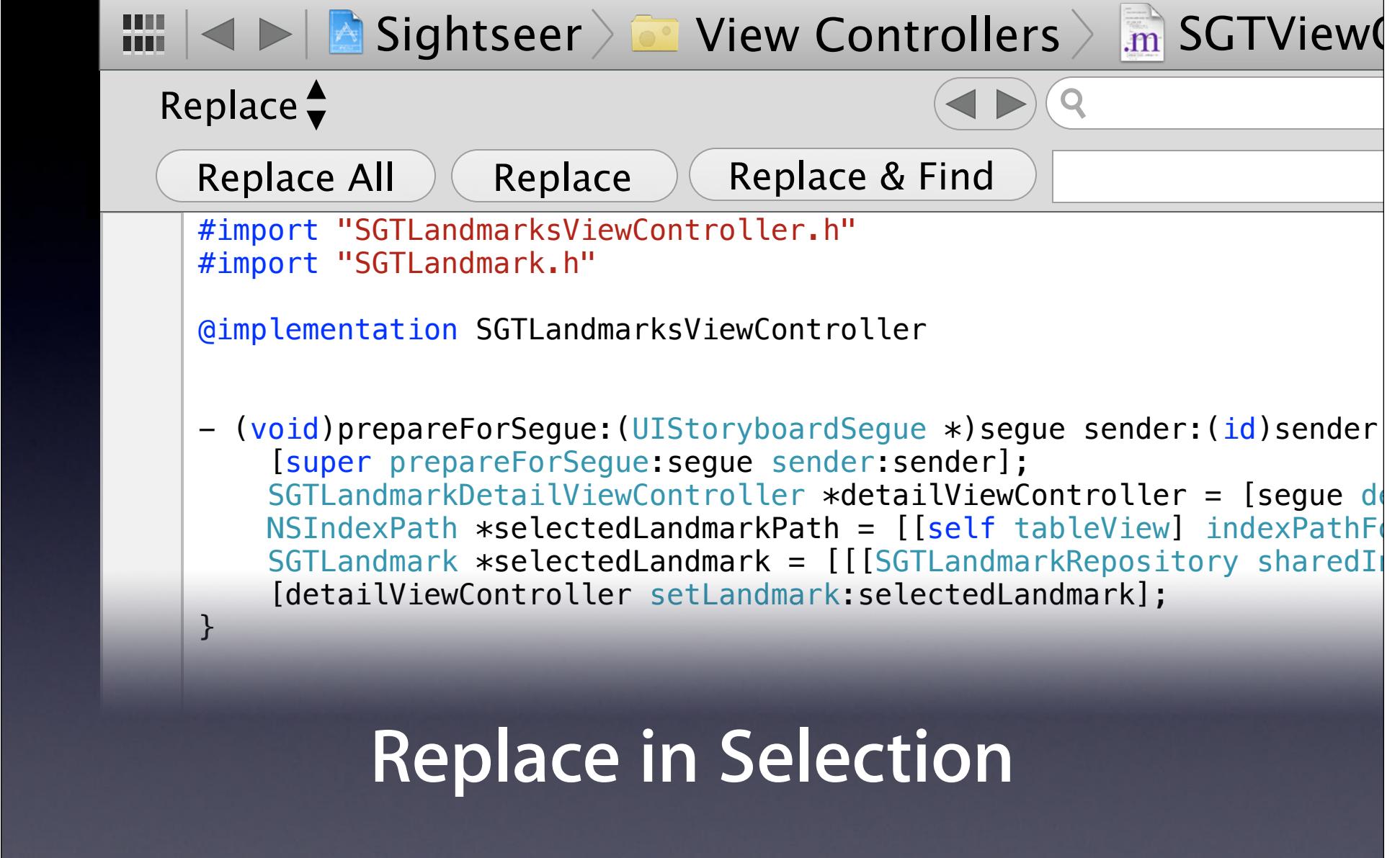
```
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    [super prepareForSegue:segue sender:sender];
    SGTLandmarkDetailViewController *detailViewController = [segue destinationViewController];
    NSIndexPath *selectedLandmarkPath = [[self tableView] indexPathForSelectedRow];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

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell forRowAtIndexPath:(NSIndexPath *)indexPath
{
    [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"];
    if (!cell)
    {
        cell = [[[NSBundle mainBundle] loadNibNamed:@"SGTLandmarkTableViewCell" owner:self options:nil] objectAtIndex:0];
    }
    cell.landmark = [[[SGTLandmarkRepository sharedInstance] landmarks] objectAtIndex:indexPath.row];
    return cell;
}
```

Trailing Whitespace Removal



Sightseer > View Controllers > SGTVController.m

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] indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedRepository] landmarks] objectAtIndex:selectedLandmarkPath.row];
    [detailViewController setLandmark:selectedLandmark];
}
```

Replace in Selection

The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the "Sightseer" project with a "View Controllers" group containing "SGTViewController".
- Search Bar:** Displays "Replace" and ".m" file icon.
- Toolbar:** Includes "Replace All", "Replace", "Replace & Find", and "current" buttons.
- Text Editor:** Contains the following code:

```
#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 sharedRepository] landmarks] objectAtIndex:selectedLandmarkPath.row];
    [detailViewController setLandmark:selectedLandmark];
}
```

Replace in Selection

Sightseer > View Controllers > SGTVController.m

Replace

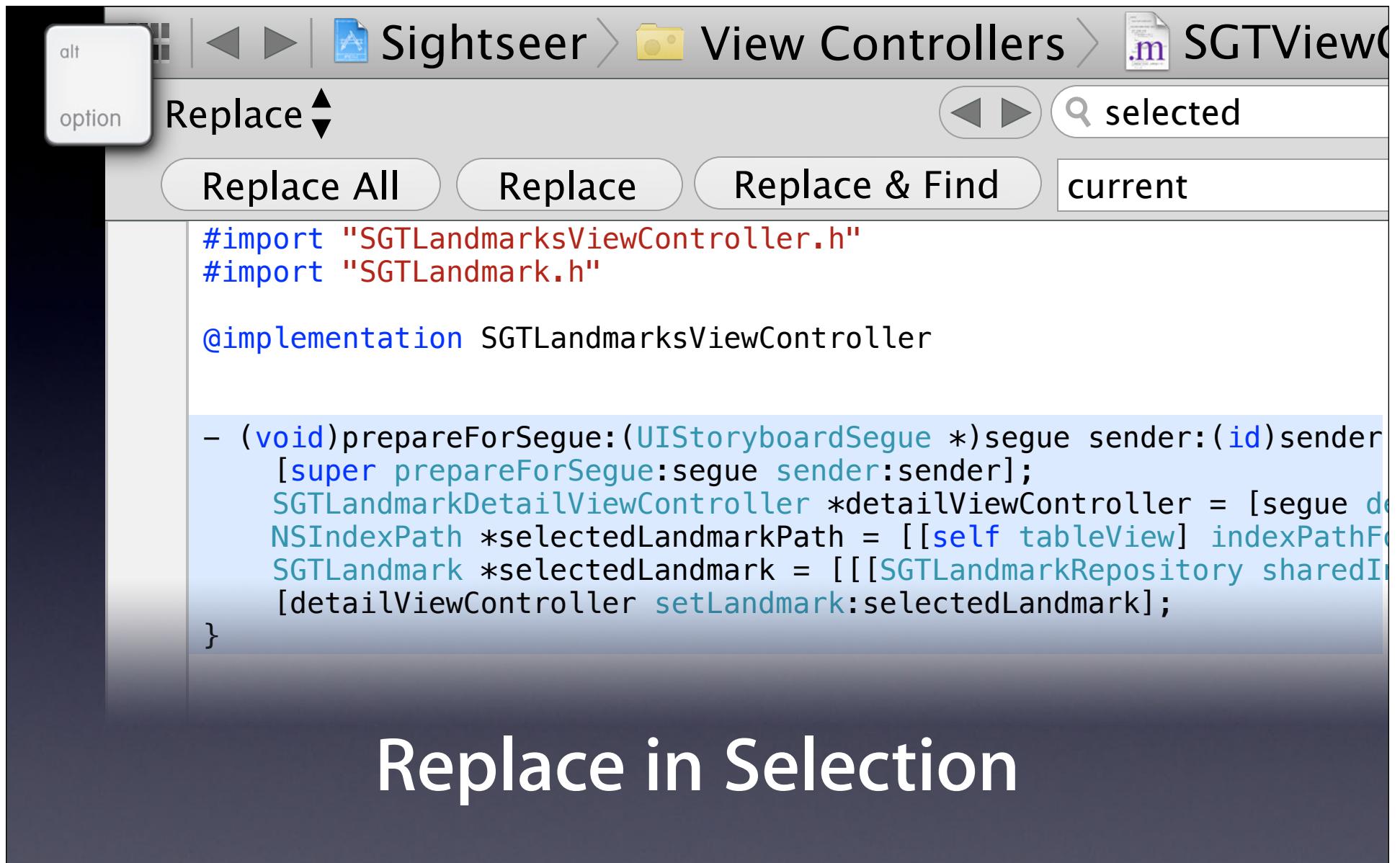
Replace All Replace Replace & Find current

```
#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 sharedRepository] landmarks] objectAtIndex:selectedLandmarkPath.row];
    [detailViewController setLandmark:selectedLandmark];
}
```

Replace in Selection



The screenshot shows the Xcode interface with the Replace panel open. The title bar indicates the project is "Sightseer" and the file is "View Controllers / SGTVController.m". The Replace panel has several buttons: "Replace All", "Replace", "Replace & Find", and "current". The "Replace" button is currently selected. On the left, there are keyboard shortcut keys: "alt" and "option". The main area contains the following code:

```
#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 sharedRepository] landmarks] objectAtIndex:selectedLandmarkPath.row];
    [detailViewController setLandmark:selectedLandmark];
}
```

Below the code, the text "Replace in Selection" is displayed in large white font.

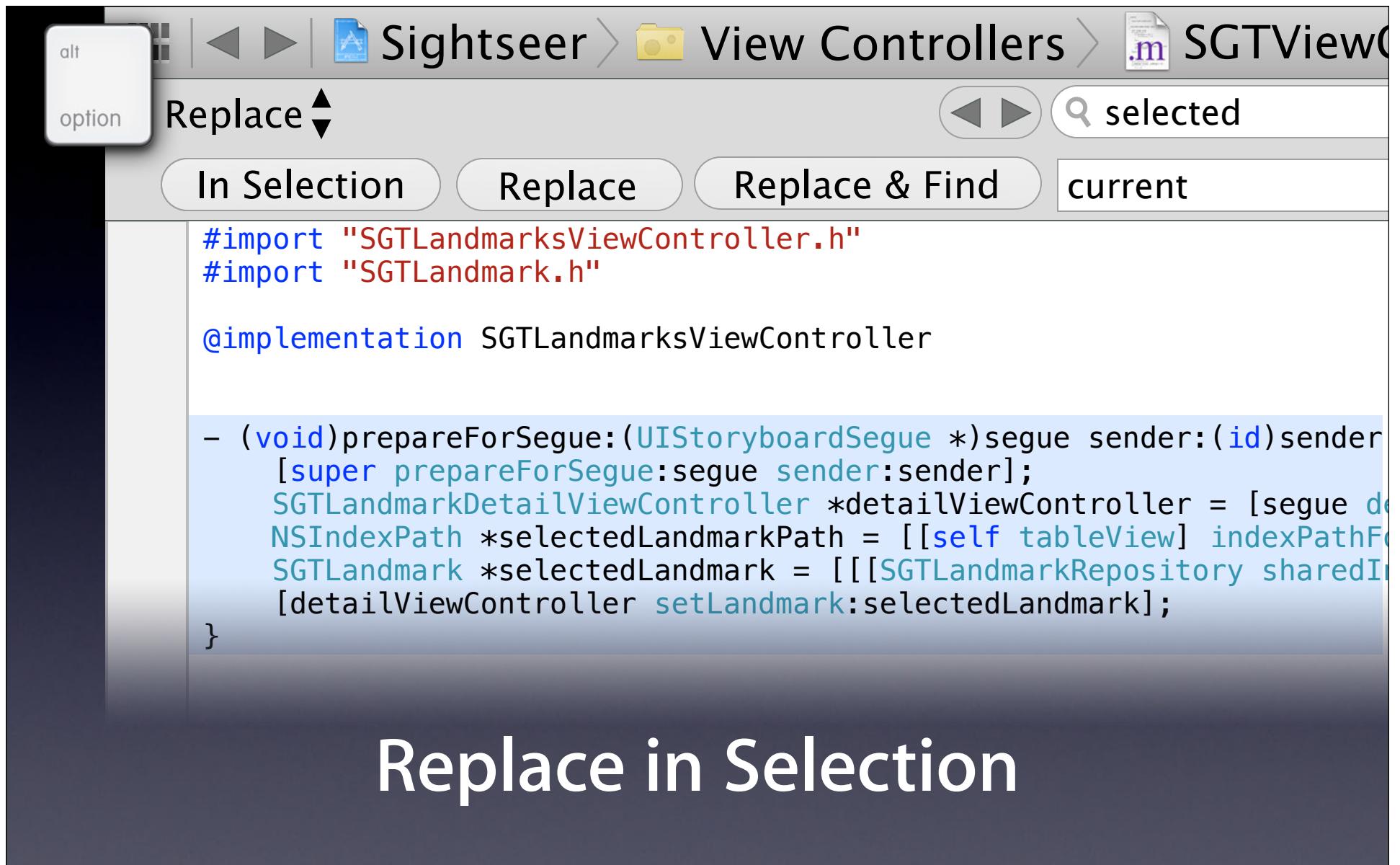
The screenshot shows the Xcode interface with the Replace tool open. The title bar indicates the project is 'Sightseer' and the file is 'View Controllers / SGTV...'. The Replace bar at the top has several buttons: 'Replace' (with a dropdown arrow), 'selected' (with arrows), 'Replace All' (highlighted with a yellow box), 'Replace', 'Replace & Find', and 'current'. Below the bar is a code editor window displaying Objective-C code for 'SGTLandmarksViewController.m'. The code includes imports for 'SGTLandmarksViewController.h' and 'SGTLandmark.h', an implementation block, and a method implementation for 'prepareForSegue:sender:'.

```
#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 sharedRepository] landmarks] objectAtIndex:selectedLandmarkPath.row];
    [detailViewController setLandmark:selectedLandmark];
}
```

Replace in Selection



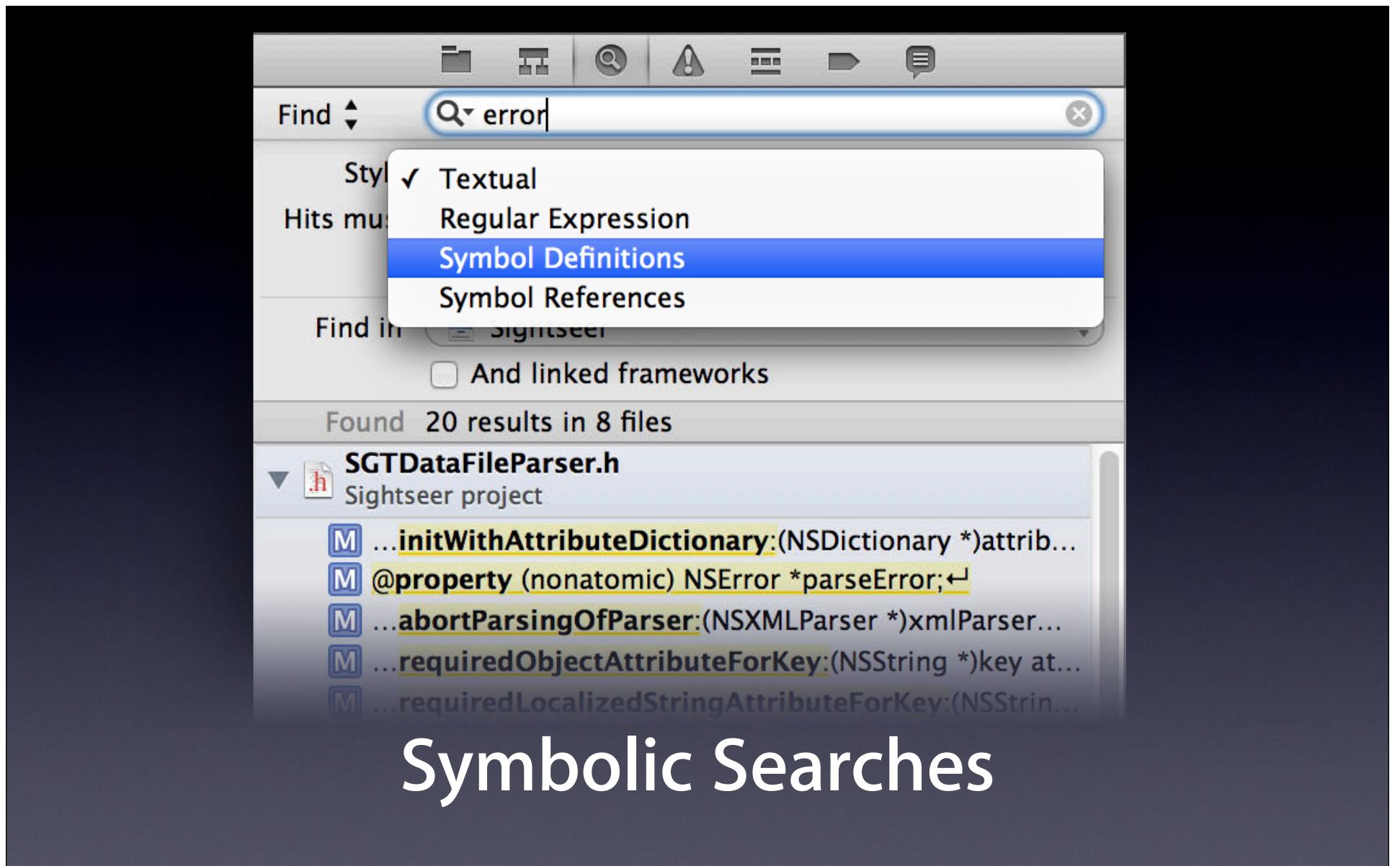
The screenshot shows the Xcode interface with the Replace tool open. The title bar indicates the project is "Sightseer" and the file is "View Controllers / SGTVController.m". The Replace interface has several buttons: "Replace" (highlighted), "In Selection", "Replace & Find", and "current". There are also navigation buttons for previous and next replacements, and a search bar labeled "selected". A key binding "option" is shown above the Replace button. The code editor displays the implementation of the `SGTLandmarksViewController` class, specifically the `- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender` method. The placeholder text "Replace in Selection" is overlaid at the bottom of the code editor.

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

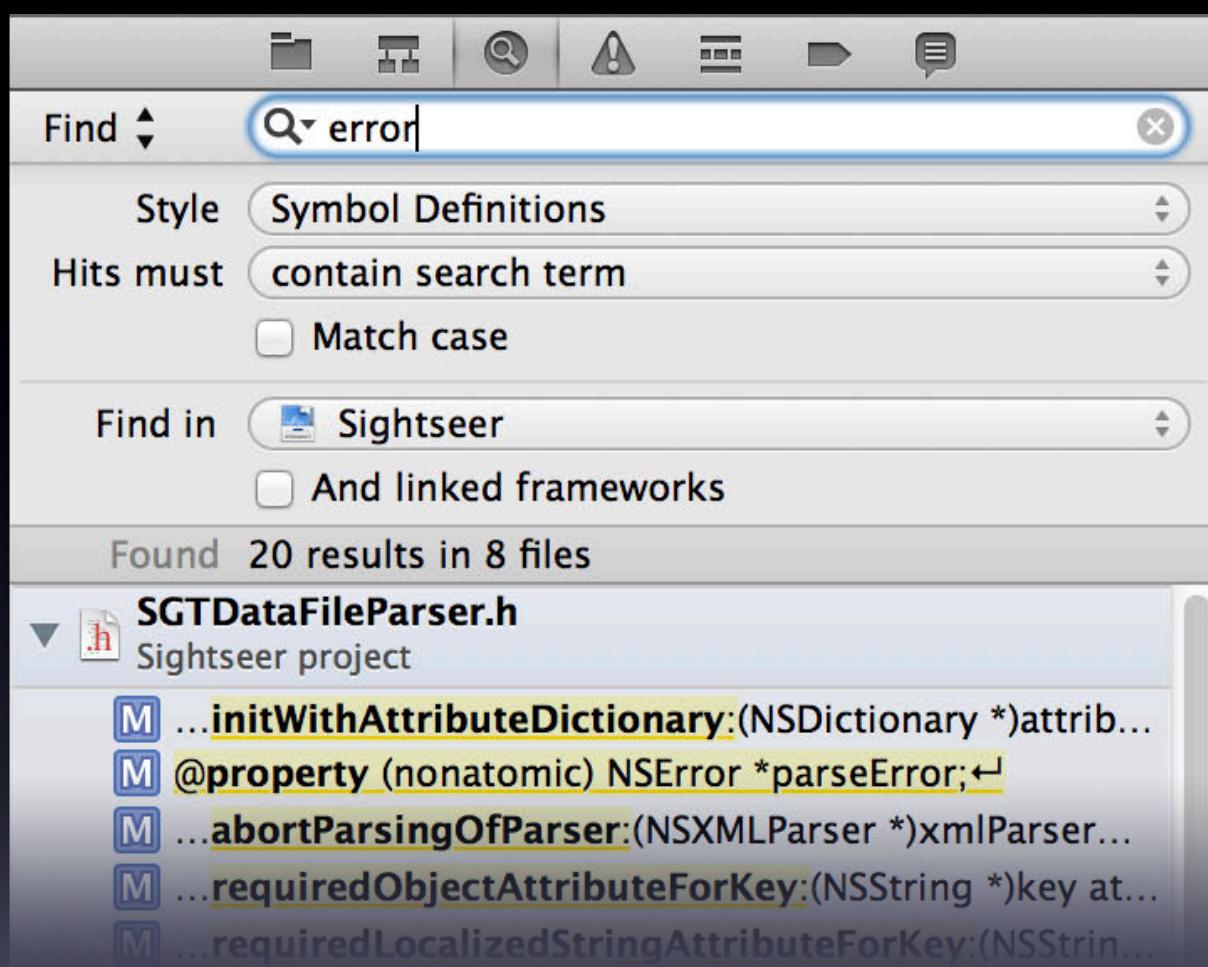
@implementation SGTLandmarksViewController

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

Replace in Selection



Symbolic Searches



Search Performance

grep

Xcode

19.3 seconds

grep

Xcode

19.3 seconds

grep

4.6 seconds

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 sharedRepository] sharedRepository] objectAtIndex:selectedLandmarkPath.row];
    [detailViewController setLandmark:selectedLandmark];
}

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

Caller and Callee Assistant Editor

<pre> #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:(NSInteger)section return [[[SGTLandmarkRepository sharedInstance] landmarks] count]; } - (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPathAtIndexPath:(NSIndexPath SGTLandmarkTableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:@"La SGTLandmark *landmark = [[[SGTLandmarkRepository sharedInstance] landmarks] object [cell nameLabel] setText:[landmark localizedName]]; return cell; } @end </pre>	<pre> #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:(NSInteger)section return [[[SGTLandmarkRepository sharedInstance] landmarks] count]; } - (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPathAtIndexPath:(NSIndexPath SGTLandmarkTableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:@"La SGTLandmark *landmark = [[[SGTLandmarkRepository sharedInstance] landmarks] object [cell nameLabel] setText:[landmark localizedName]]; return cell; } @end </pre>
--	--

Selective Commit

<pre> #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:(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 </pre>	✓ 1 ▾	<pre> #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:(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 </pre>
---	--	---

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] objectAtIndex:selectedLandmarkPath];
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell forRowAtIndexPath:(NSIndexPath *)indexPath {
    [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:@"LandmarkCell"];
    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 destinationViewController];
    NSIndexPath *selectedLandmarkPath = [[self tableView] indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance] landmarks] objectAtIndex:selectedLandmarkPath];
    [detailViewController setLandmark:selectedLandmark];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell forRowAtIndexPath:(NSIndexPath *)indexPath {
    [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:@"LandmarkCell"];
    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 destinationViewController];
    NSIndexPath *selectedLandmarkPath = [[self tableView] indexPathForSelectedRow];
    SGTLandmark *selectedLandmark = [[[SGTLandmarkRepository sharedInstance] landmarks]
        [detailViewController setLandmark:selectedLandmark]];
}

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell forRowAtIndexPath:(NSIndexPath *)indexPath {
    [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:@"LandmarkCell"];
    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 forRowAtIndexPath:(NSIndexPath *)indexPath {
    [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:@"LandmarkCell"];
    SGTLandmark *landmark = [[[SGTLandmarkRepository sharedInstance] landmarks] objectAtIndex:indexPath.row];
    [cell nameLabel setText:[landmark localizedName]];
    return cell;
}
```



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 forRowAtIndexPath:(NSIndexPath *)indexPath {
    [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:@"LandmarkCell"];
    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 forRowAtIndexPath:(NSIndexPath *)indexPath {
    [cell setBackgroundColor:[UIColor clearColor]];
}

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

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



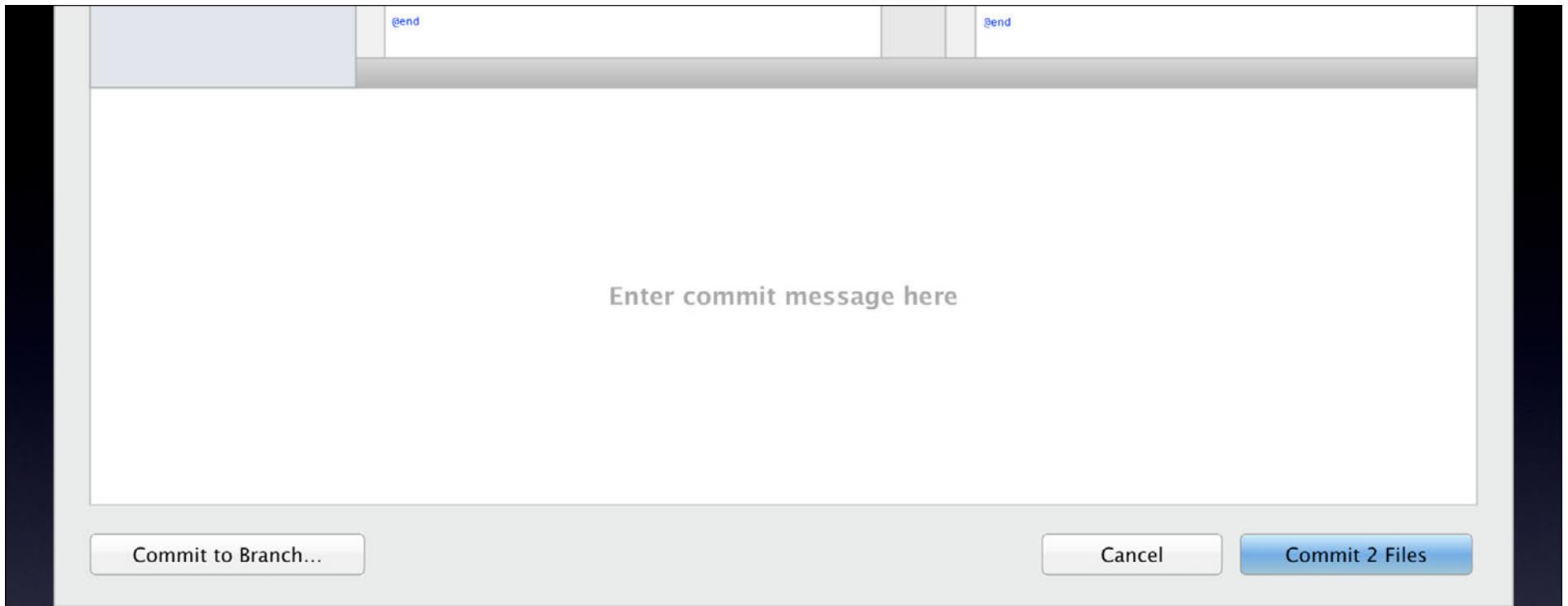
1



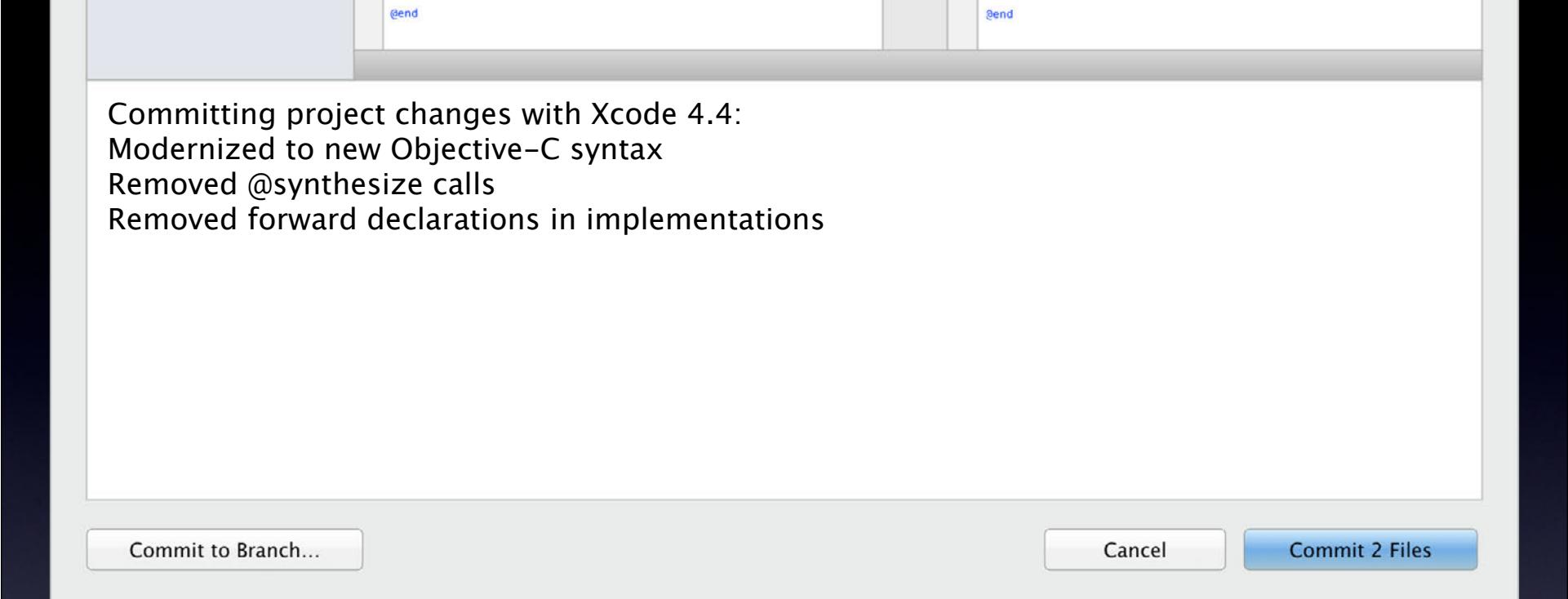
Don't Commit

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

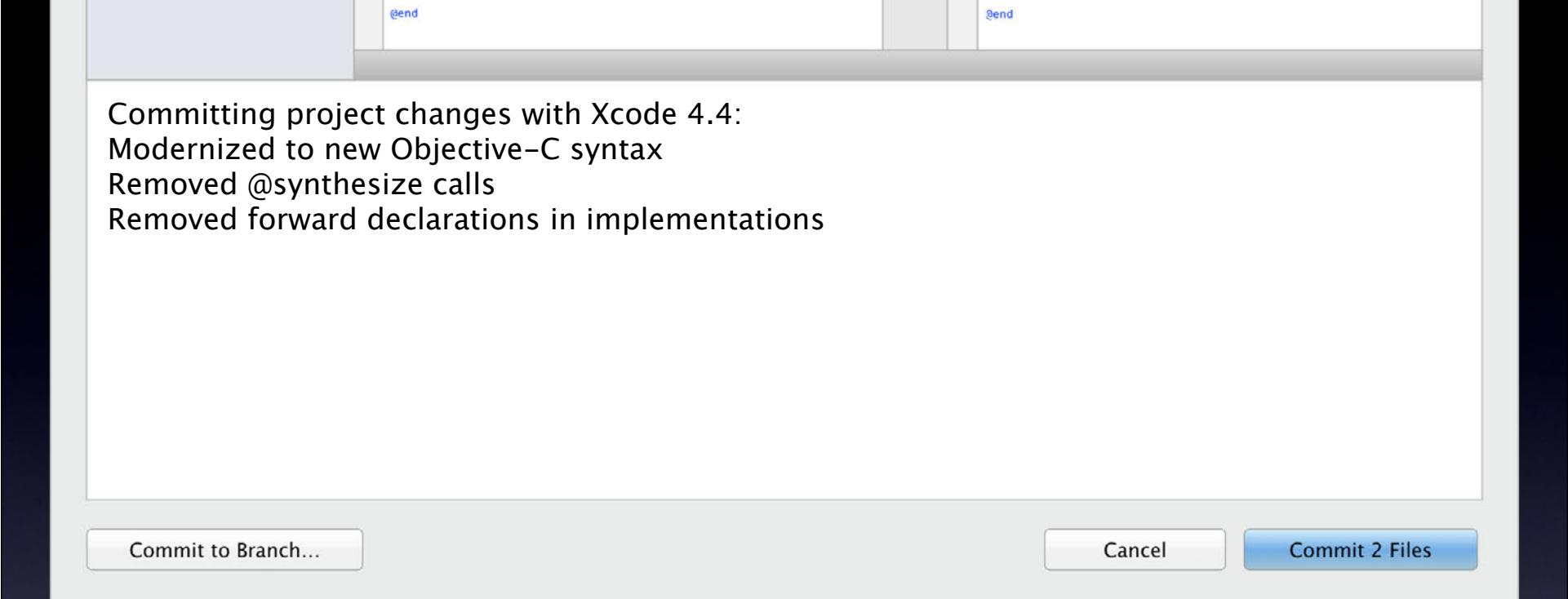
Committing project changes with Xcode 4.4:
Modernized to new Objective-C syntax
Removed @synthesize calls
Removed forward declarations in implementations

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

EdIt

Design

Design

I

Integrated

Sightseer / Sightseer / View Controllers / SGTLandmarkSplitViewController.m

Build Sightseer: Succeeded | Yesterday at 6:51 PM

```

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

@interface SGTLandmarkSplitViewController () <NSTableViewDataSource, NSTableViewDelegate>
@property (weak, nonatomic) IBOutlet NSTableView *tableView;
@property (weak, nonatomic) IBOutlet NSView *detailAreaContentView;
@property (nonatomic) NSViewController <SGTLandmarkSplitViewControllerDelegate> *installedDetailViewController;
@end

@implementation SGTLandmarkSplitViewController {
    NSArray *_landmarks;
}

@synthesize tableView = _tableView;
@synthesize detailAreaContentView = _detailAreaContentView;

- (void)setDetailViewControllerClass:(Class)class {
    if ([class isSubclassOfClass:[SGTLandmarkSplitViewController class]]) {
        self.detailViewControllerClass = class;
        [self reinstallDetailViewController];
    }
}

- (void)loadView {
    // If we've already loaded the view, so that the first load grabs this data
    [[SGTLandmarkRepository sharedInstance] registerLandmarkDidChangeHandler:^{
        _landmarks = [[SGTLandmarkRepository sharedInstance] landmarks];
        [[self tableView] reloadData];
    }];
    _landmarks = [[SGTLandmarkRepository sharedInstance] landmarks];
}

- (void)reinstallDetailViewController {
    if (_installedDetailViewController) {
        [_installedDetailViewController.view removeFromSuperview];
        _installedDetailViewController = nil;
    }

    Class detailViewControllerClass = [self detailViewControllerClass];
    NSViewController <SGTLandmarkSplitViewControllerDelegate> *viewController = ([self detailAreaContentView] ? [[detailViewControllerClass alloc] sgt_initUsingNibName] : nil);
    if (viewController) {
        viewController.view = [viewController view];
        [viewController setTranslatesAutoresizingMaskIntoConstraints:NO];
        [[self detailAreaContentView] addSubview:viewController.view];
        [[self detailAreaContentView] addConstraints:[NSLayoutConstraint constraintsWithVisualFormat:@"H:[view]" options:0 metrics:nil views:NSDictionaryOfVariableBindings(view)]];
        [[self detailAreaContentView] addConstraints:[NSLayoutConstraint constraintsWithVisualFormat:@"V:[view]" options:0 metrics:nil views:NSDictionaryOfVariableBindings(view)]];
    }
    _installedDetailViewController = viewController;
}

- (void)sgt_viewDidLoadInstall {
    NSLog(@"%@", __PRETTY_FUNCTION__);
}

- (void)sgt_viewWillUninstall {
    NSLog(@"%@", __PRETTY_FUNCTION__);
}

#pragma mark - Landmark Data Source
#pragma mark -
#pragma mark -
- (NSInteger)numberOfRowsInTableView:(NSTableView *)tableView {
    return [_landmarks count];
}

- (id)tableView:(NSTableView *)tableView objectValueForTableColumn:(NSTableColumn *)tableColumn row:(NSInteger)row {
    return [_landmarks objectAtIndex:row];
}

- (void)tableViewSelectionDidChange:(NSNotification *)notification {
}

```

Identity and Type

File Name: SGTLandmarkSplitViewController.m

File Type: Default – Objective-C source code

Location: Relative to Group

SGTLandmarkSplitViewController.m

Full Path: /Volumes/Documents/Applications/DeveloperTools/Events/2012.09 -- WWDC/WWDC2012/SharedDemoApps/Sightseer/OS X/ViewControllers/SGTLandmarkSplitViewController.m

Localization

Target Membership

Text Settings

Source Control

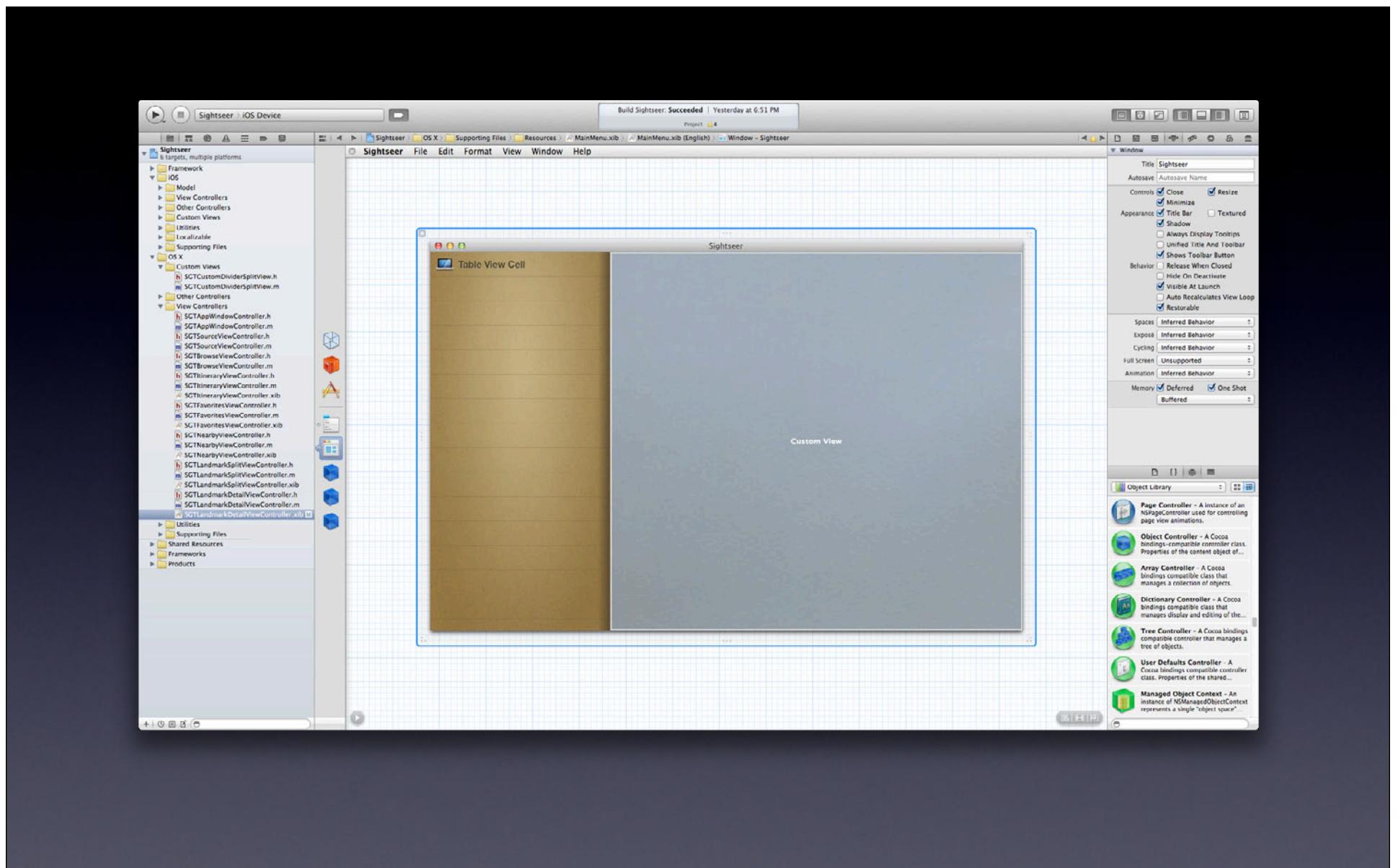
Version --

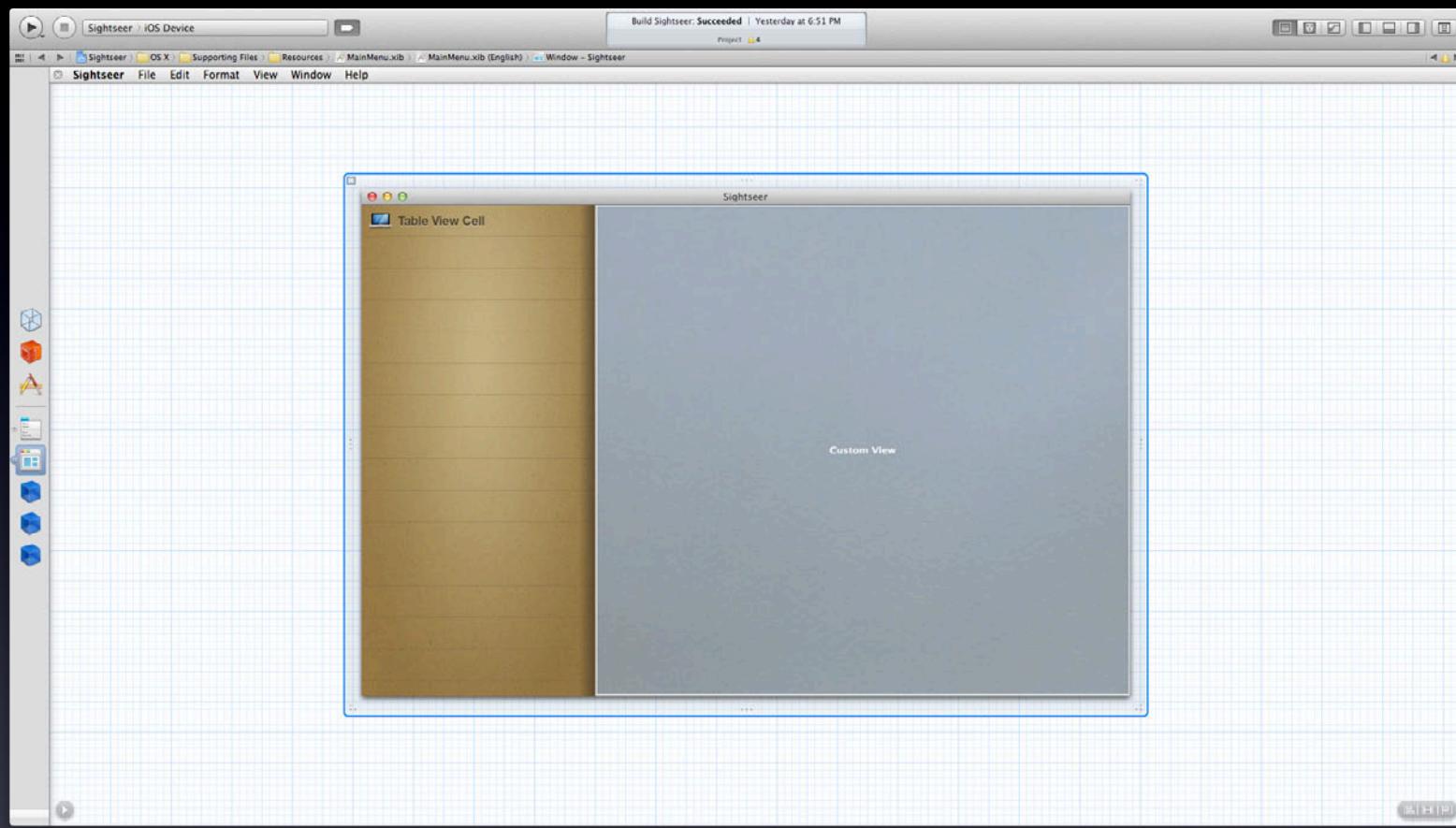
Status: No changes

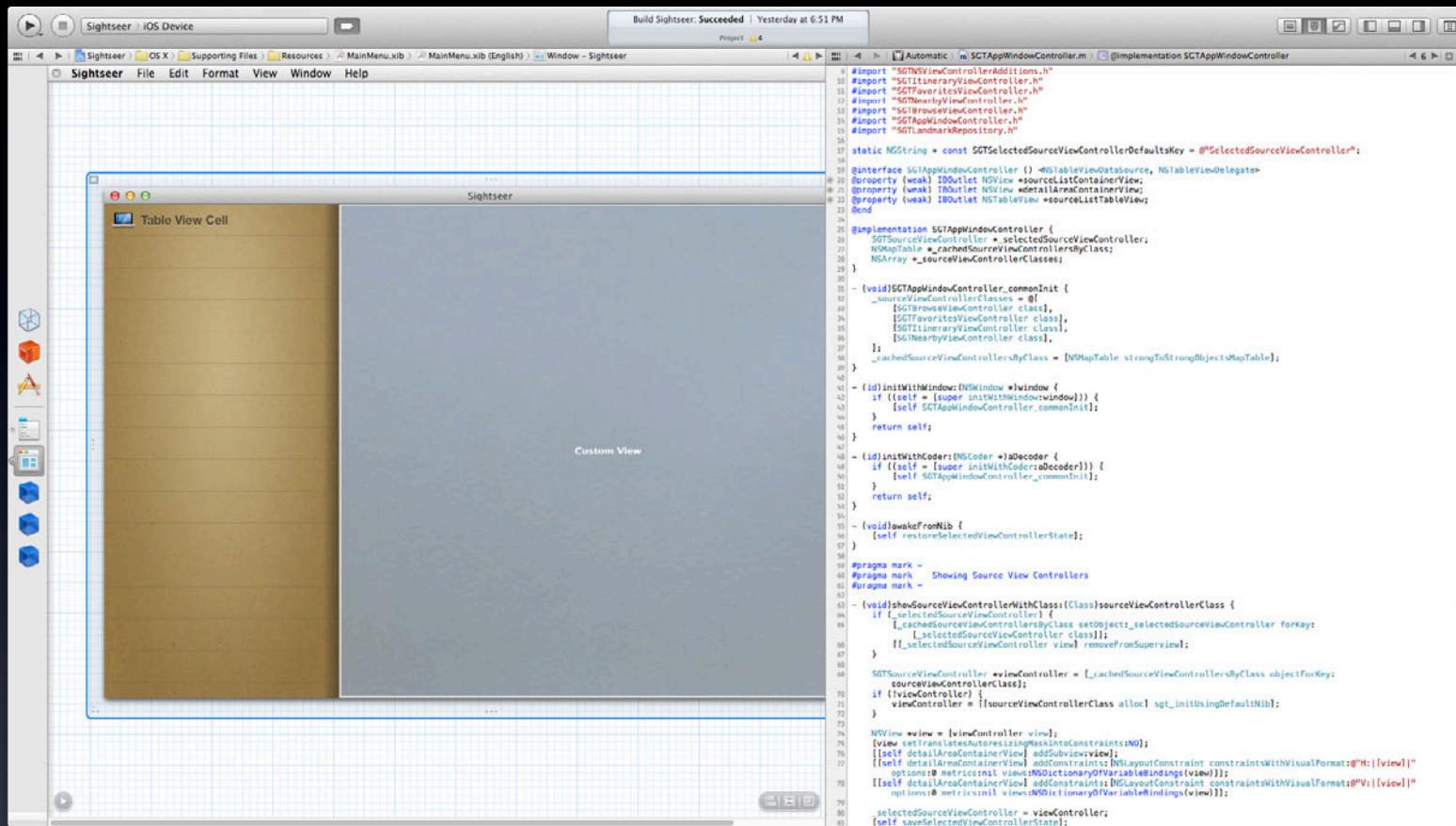
Location: svn+ssh://svn.apple.com/svn/wwdcbox/devtech/WWDC2012/SharedDemoApps/Sightseer/OS X/ViewControllers/SGTLandmarkSplitViewController.m

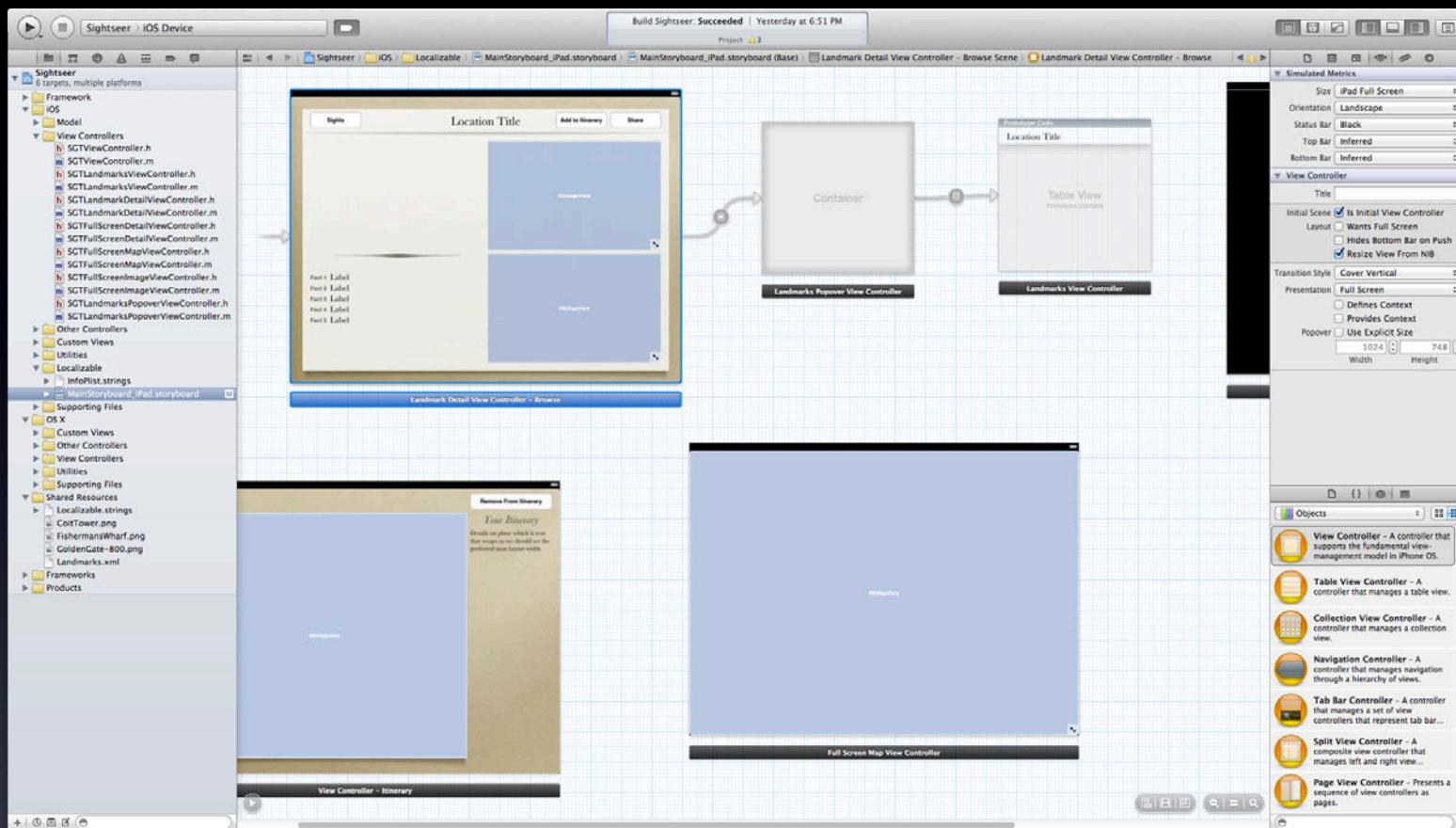
Object Library

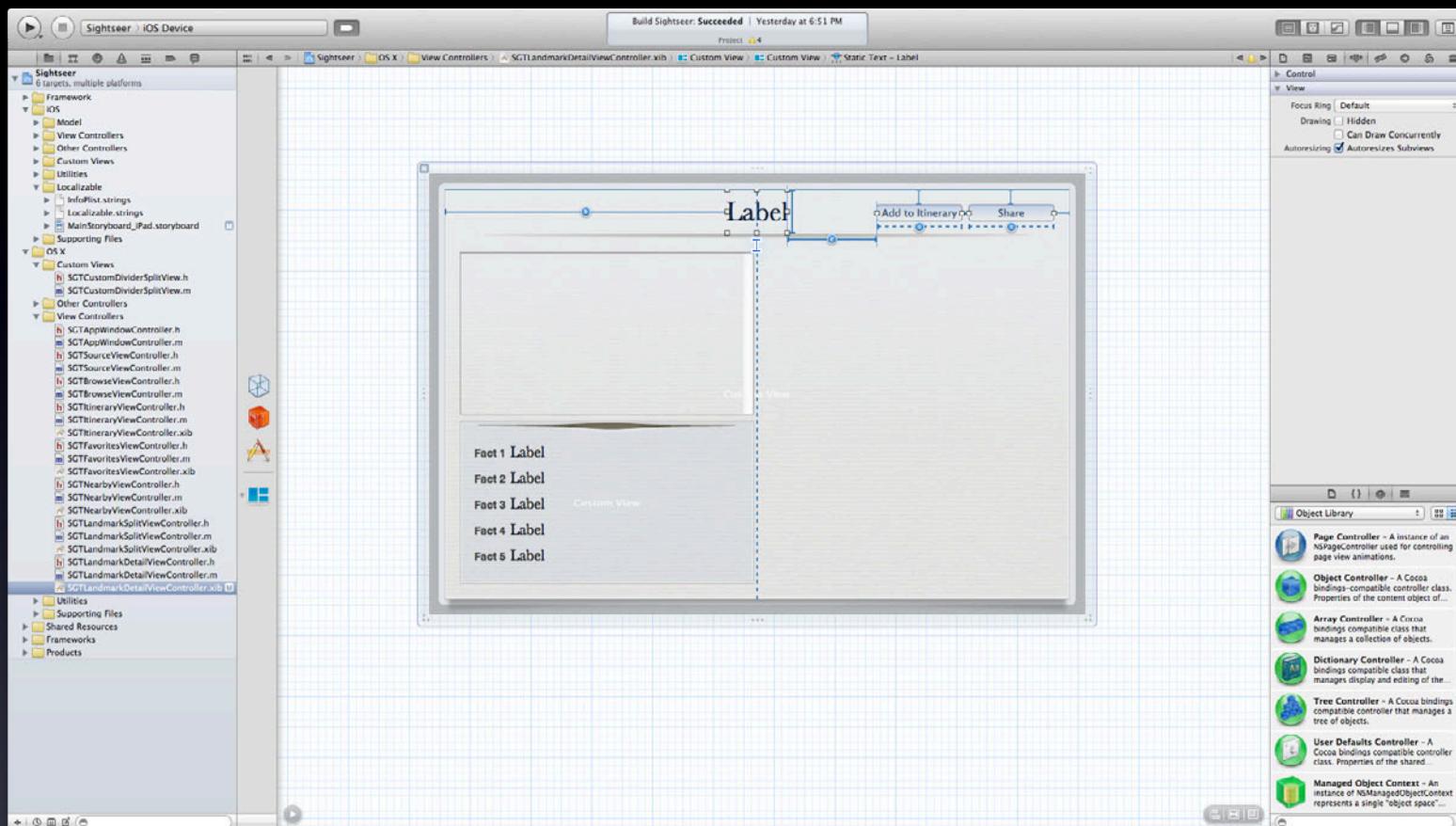
- Page Controller – A instance of an NSPageController used for controlling page view animations.
- Object Controller – A Cocoa bindings-compatible controller class. Properties of the content object of...
- Array Controller – A Cocoa bindings-compatible class that manages a collection of objects.
- Dictionary Controller – A Cocoa bindings-compatible class that manages display and editing of the...
- Tree Controller – A Cocoa bindings-compatible controller that manages a tree of objects.
- User Defaults Controller – A Cocoa bindings-compatible controller class. Properties of the shared...
- Managed Object Context – An instance of NSManagedObjectContext represents a single “object space”.

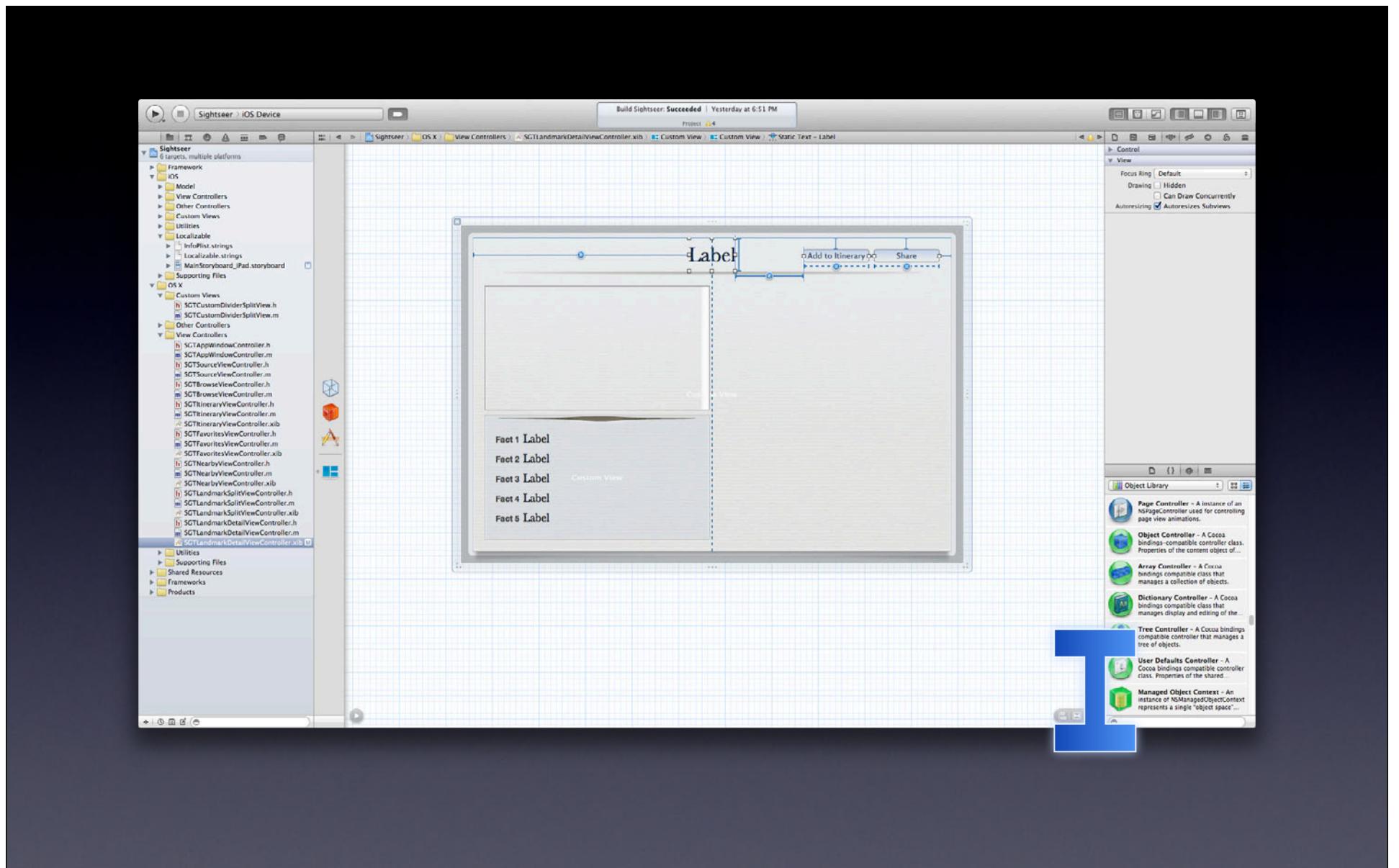


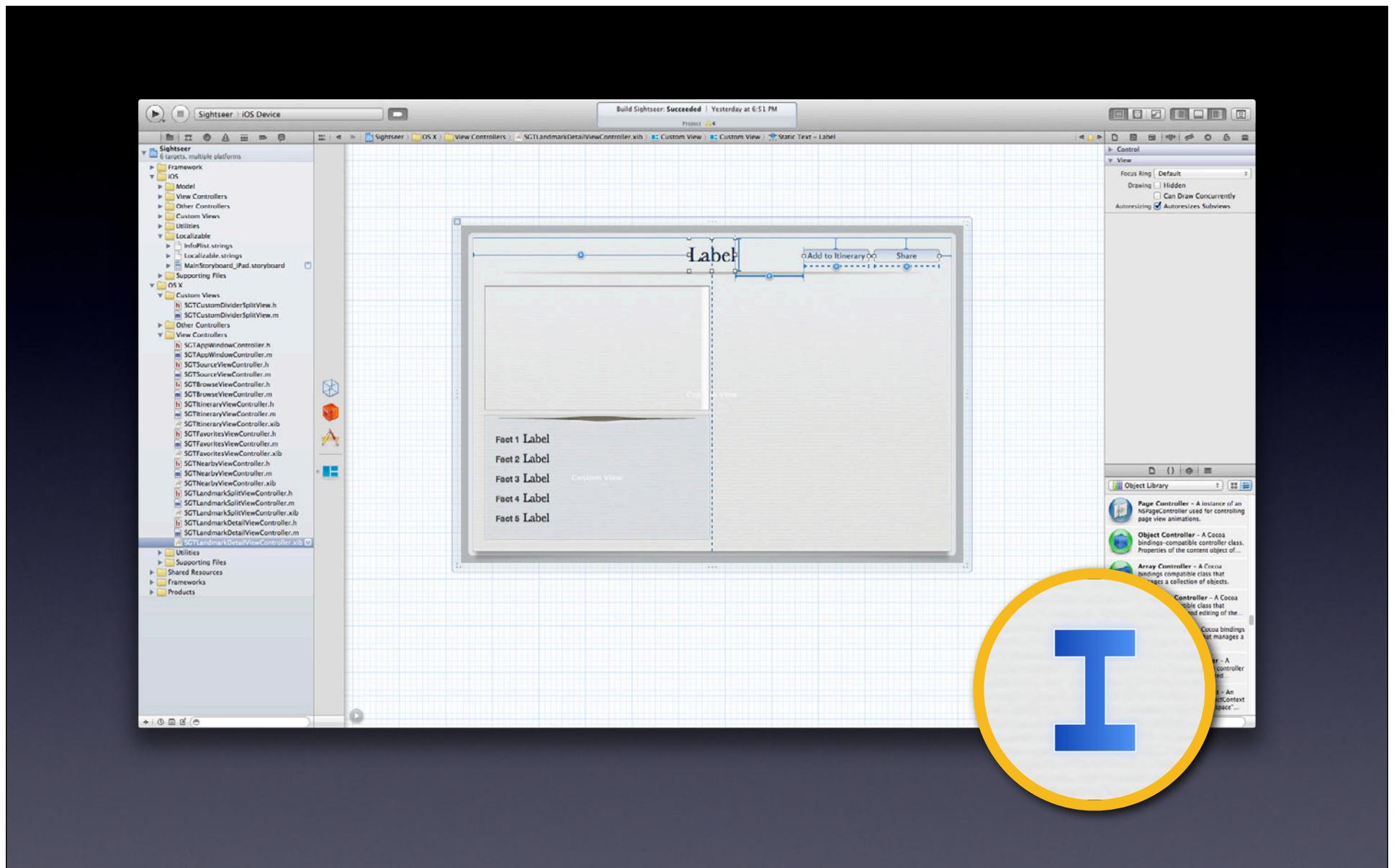


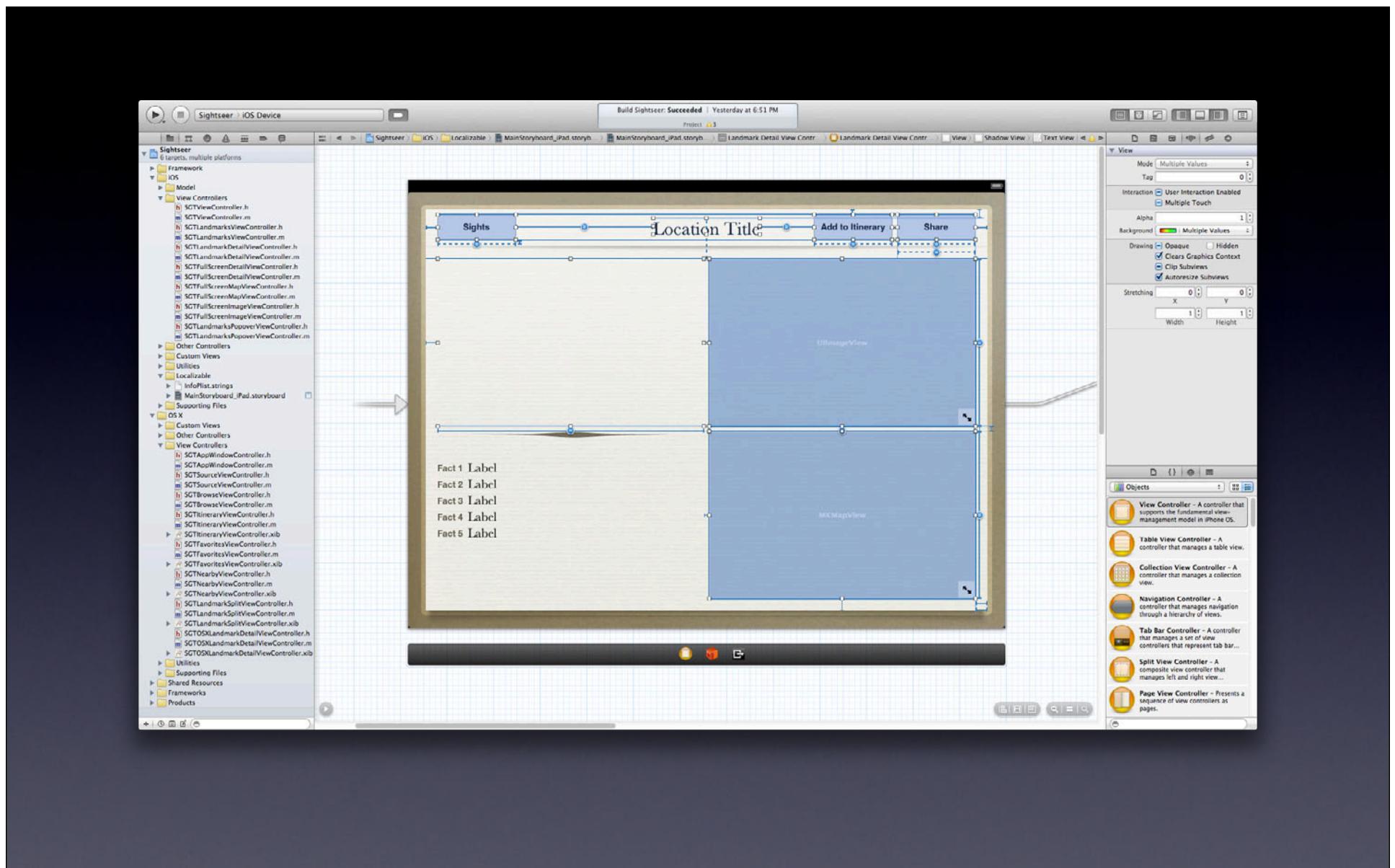


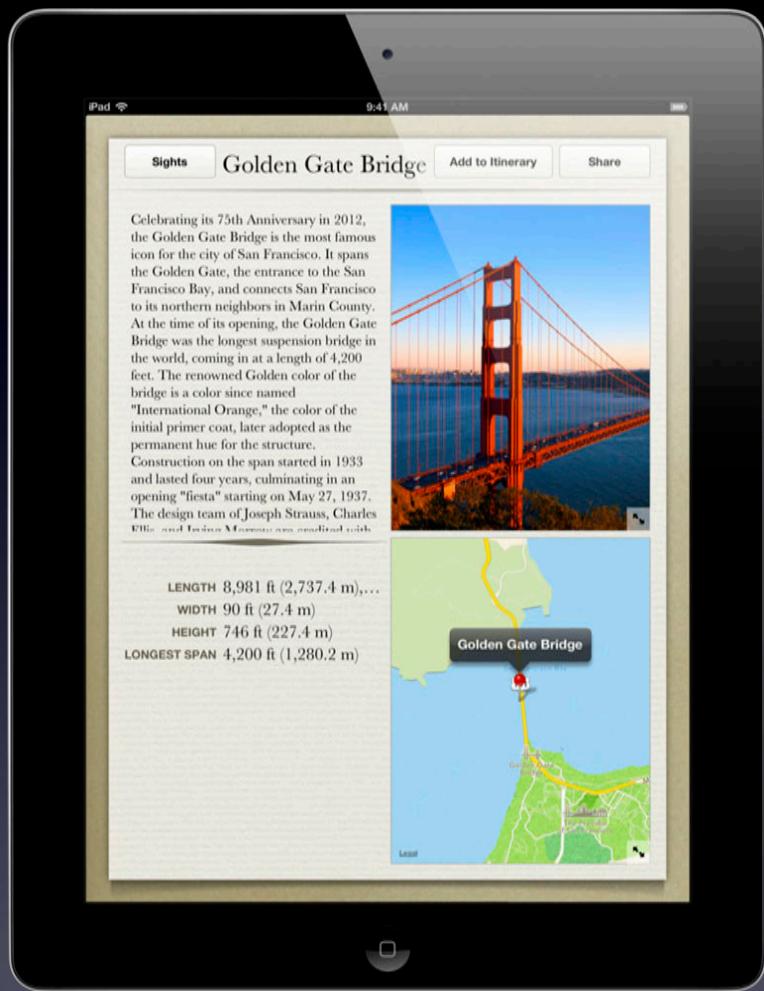




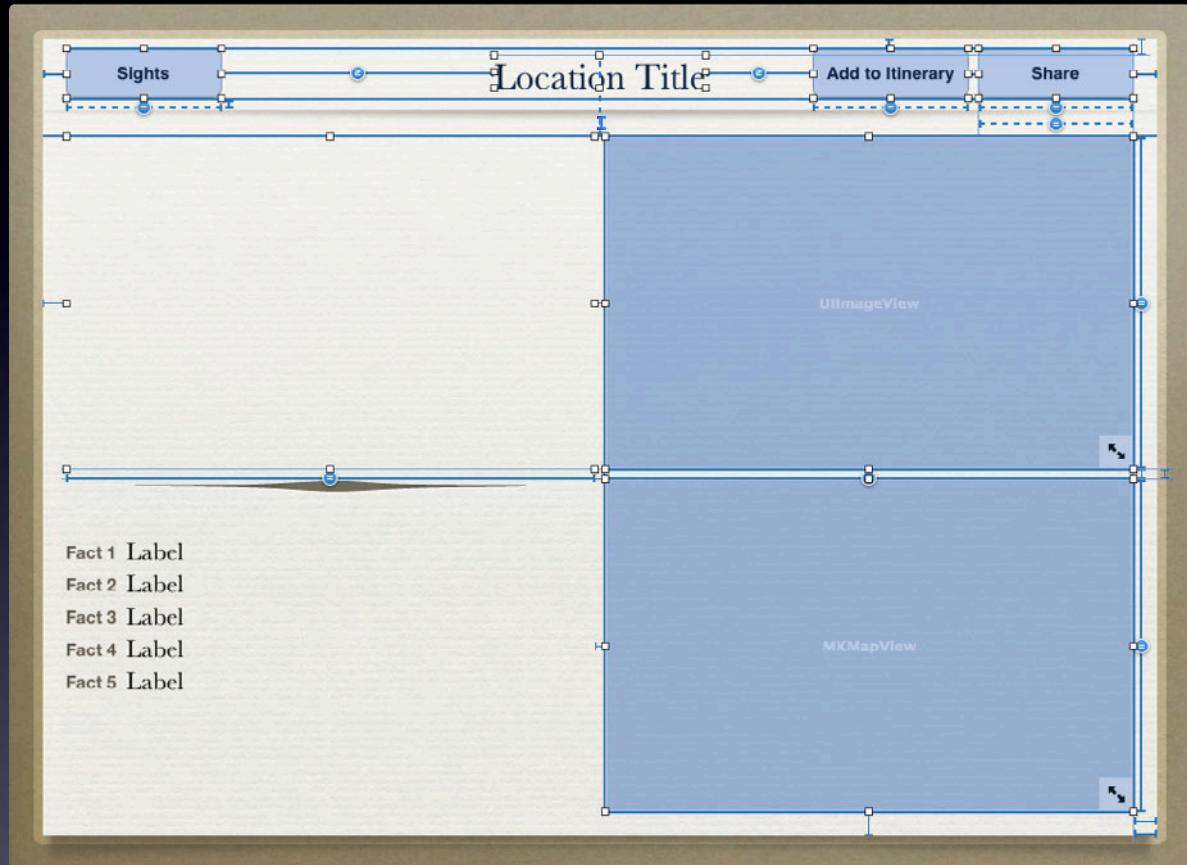












Localize



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 numberOfRowsInSection:(NSInteger)section
{
    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.row];
    [detailViewController setLandmark:selectedLandmark];
}

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

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

@implementation SGTLandmarksViewController

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
    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.row];
    [detailViewController setLandmark:selectedLandmark];
}

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

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

@implementation SGTLandmarksViewController

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
    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.row];
    [detailViewController setLandmark:selectedLandmark];
}

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



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

@implementation SGTLandmarksViewController

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

⚠ Issue

⚠ Unknown type name “SGTLandmar”: did you mean “SGTLandmark”?

Fix-it

Replace “SGTLandmar” with “SGTLandmark”

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

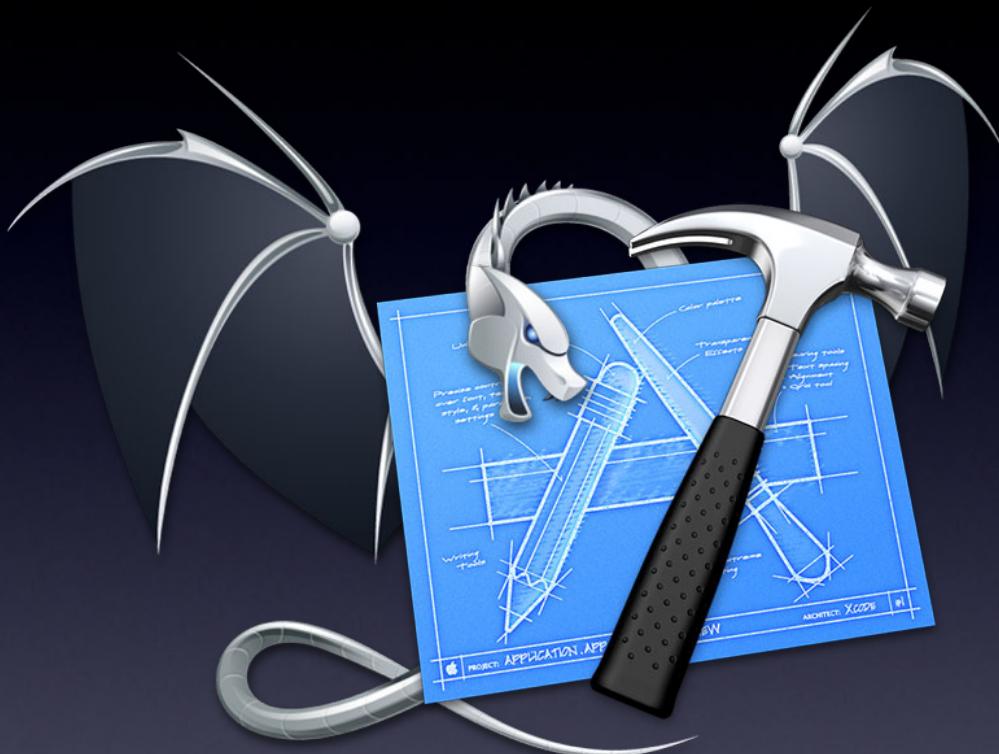
- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell forRowAtIndexPath:(NSIndexPath *)indexPath
{
    [cell setBackgroundColor:[UIColor clearColor]];
}
```





LLVM

Chris Lattner
Director, Development Technologies





LLDB

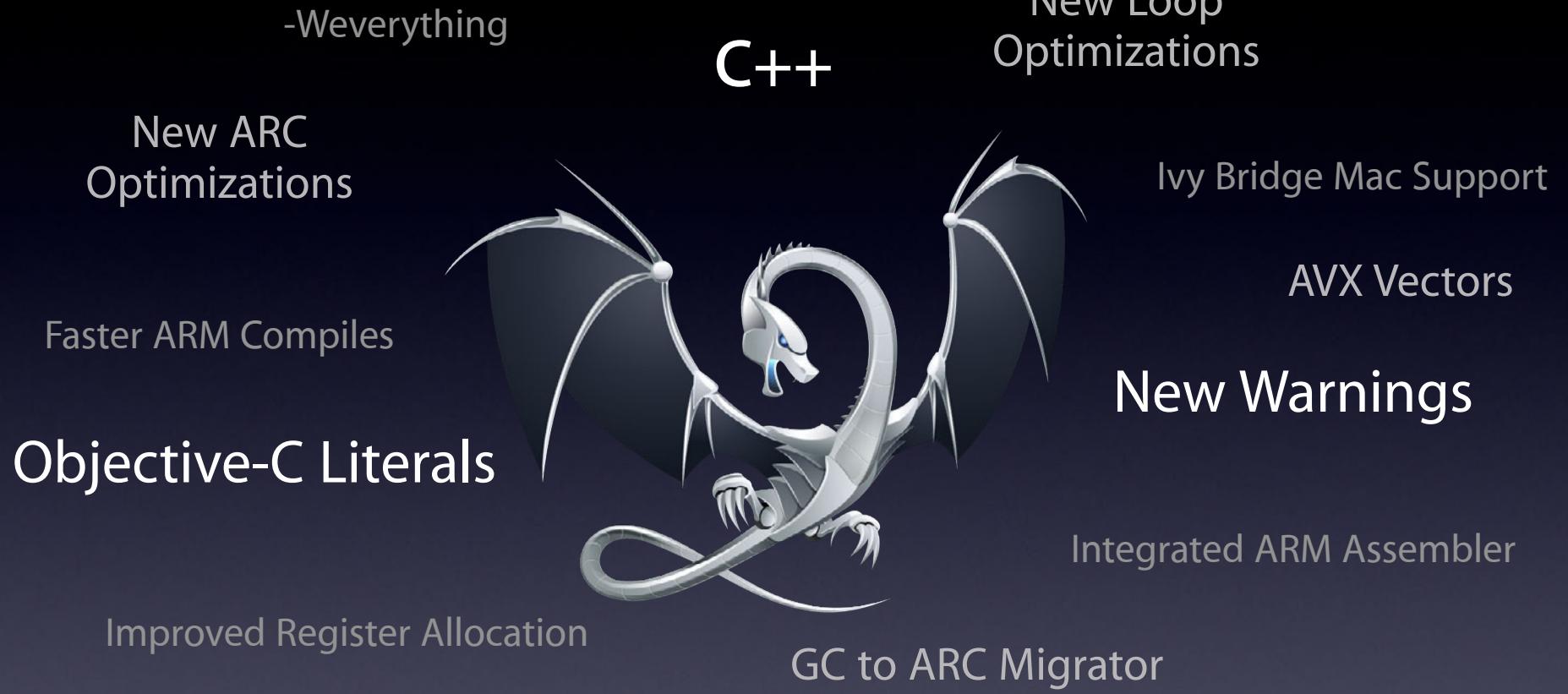


LLDB

Apple LLVM Compiler 4







C++





Xcode 4

Full C++98 Support

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, ...

Xcode 4.3

Defaulted special member functions

Delegating constructors

Explicit conversion operations

Implicitly move constructors & assignment operators

Non-static data member initializers

Raw string literals

Unicode string literals

Alignment support - alignas



Xcode 4.4

Atomics

Generalized constants - constexpr

Generalized initializer lists

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



Xcode 4.4

Atomics
Generalized constants - constexpr
Generalized initializer lists

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



Xcode 4.4

Atomics
Generalized constants - constexpr
Generalized initializer lists

λ Lambdas



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;  ↗ Undefined or garbage value returned to caller  
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;    ← 1. Variable 'Value' declared without an initial value  
71  
72     foo(&Value, 0);    ← 2. Calling 'foo'  
73  
74     return Value;    ← 5. Undefined or garbage value returned to caller  
75 }  
76  
77  
78
```

The screenshot shows a code editor with two functions: `foo` and `bar`. The `foo` function takes a pointer to an integer and a condition. If the condition is true, it sets the value to 0. The `bar` function declares an integer variable `Value` without an initial value, calls `foo` with its address and 0 as arguments, and then returns `Value`. Several annotations are present: a callout for `foo` indicates "3. Entered call from 'bar'". A callout for the declaration of `Value` indicates "1. Variable 'Value' declared without an initial value". A callout for the `foo` call indicates "2. Calling 'foo'". A callout for the `return` statement indicates "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

Basic All Combined Levels TestWWDC

Setting ▾ Static Analyzer – Checkers

Dead stores	Yes <input type="button" value="▼"/>
Misuse of GCD	Yes <input type="button" value="▼"/>
Misuse of malloc/free	Yes <input type="button" value="▼"/>

▀ Static Analyzer – Checkers – Objective-C

'@synchronized' with 'nil' mutex	Yes <input type="button" value="▼"/>
Improper handling of CFError and NSError	Yes <input type="button" value="▼"/>
Method signatures mismatch	Yes <input type="button" value="▼"/>
Misuse of CFNumberCreate	Yes <input type="button" value="▼"/>
Misuse of collections API	Yes <input type="button" value="▼"/>
Unused ivars	Yes <input type="button" value="▼"/>
Violation of 'self = [super init]' rule	Yes <input type="button" value="▼"/>
Violation of reference counting rules	Yes <input type="button" value="▼"/>

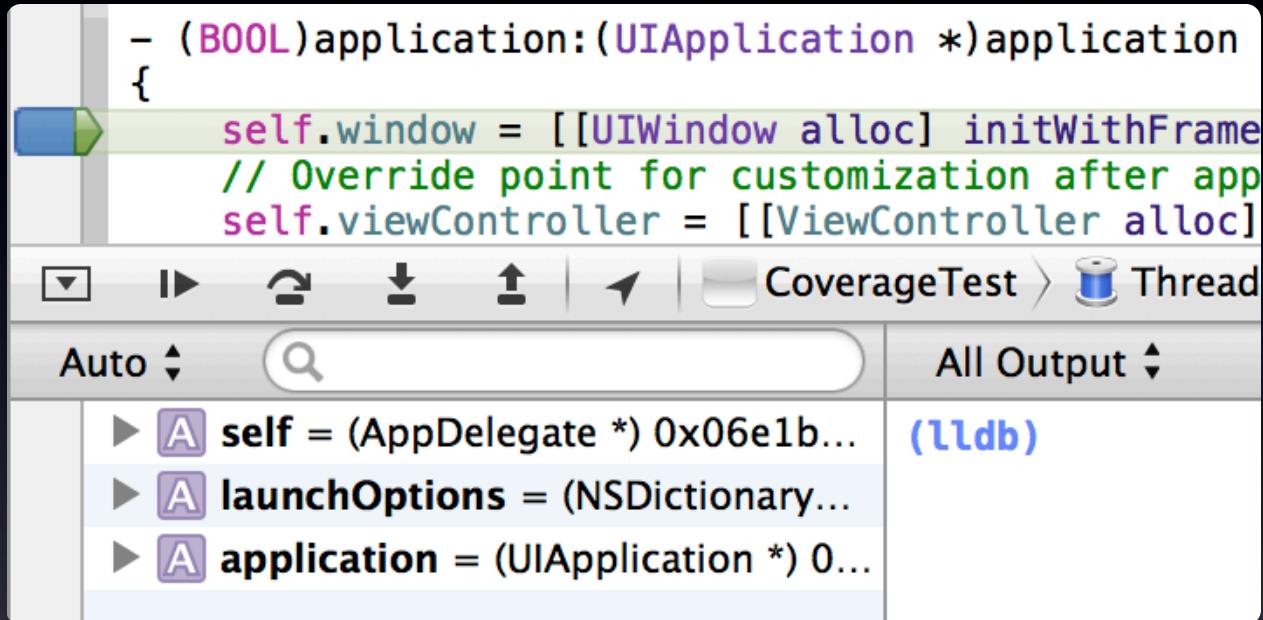
▀ Static Analyzer – Checkers – Security

Floating point value used as loop counter	No <input type="button" value="▼"/>
Misuse of Keychain Services API	Yes <input type="button" value="▼"/>
Unchecked return values	Yes <input type="button" value="▼"/>
Use of 'getpw', 'gets' (buffer overflow)	Yes <input type="button" value="▼"/>
Use of 'mktemp' or predictable 'mktemps'	Yes <input type="button" value="▼"/>
Use of 'rand' functions	No <input type="button" value="▼"/>
Use of 'strcpy' and 'strcat'	No <input type="button" value="▼"/>
Use of 'vfork'	Yes <input type="button" value="▼"/>





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



The screenshot shows the Xcode debugger window with a stack trace. The stack trace table has two columns: 'Variables' and '(lldb)'. The 'Variables' column lists local variables with their types and memory addresses. The '(lldb)' column contains the command prompt for further interaction.

Variables	(lldb)
▶ A self = (AppDelegate *) 0x06e1b...	(lldb)
▶ A launchOptions = (NSDictionary...	
▶ A application = (UIApplication *) 0...	

A screenshot of the Xcode debugger interface. The top part shows a stack trace with the following code snippet:

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

The stack trace includes the following entries:

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

The bottom part of the interface shows the LLDB command line interface with the following text:

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

(lldb) █

For more information on any particular command, try
'help <command-name>'.

(lldb) █

```
dmarkParser initWithAttributeDictionary:error:]
```

```
All Output ▾
```

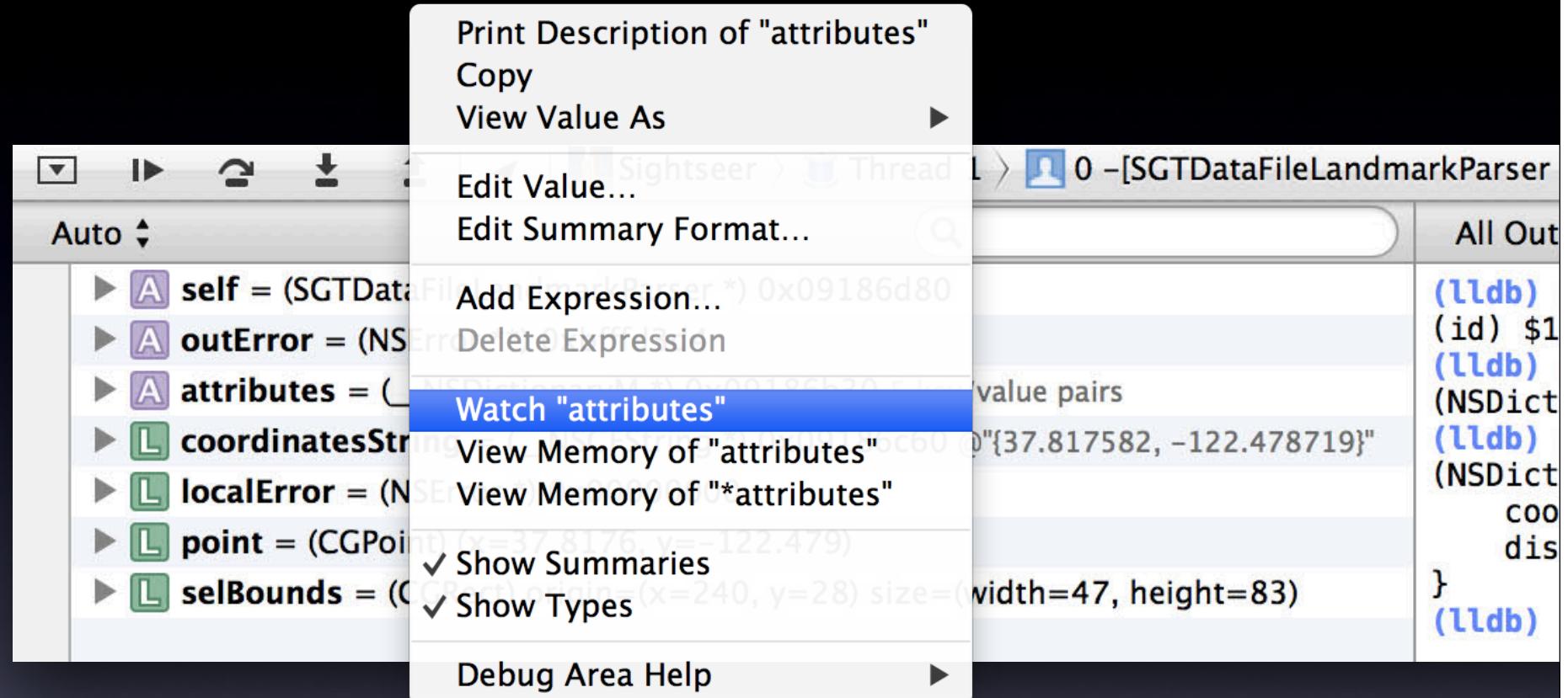
```
Clear
```

```
(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

► A	self = (SGTDataFileLandmarkParser *) 0x09186d80	(lldb)
► A	outError = (NSError **) 0xbffd3d4	(id) \$1 (lldb)
► A	attributes = (__NSDictionaryM *) 0x09186b30 5 key/value pairs	(NSDict (lldb))
► L	coordinatesString = (__NSCFString *) 0x09186c60 @{@"37.817582, -122.478719"}	(NSDict (lldb))
► L	localError = (NSError *) 0x00000000	(NSDict coo
► L	point = (CGPoint) (x=37.8176, y=-122.479)	dis
► L	selBounds = (CGRect) origin=(x=240, y=28) size=(width=47, height=83)	}







LLVM-GCC

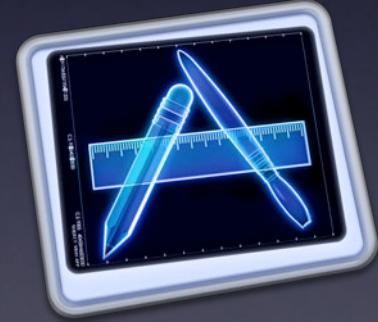


GDB





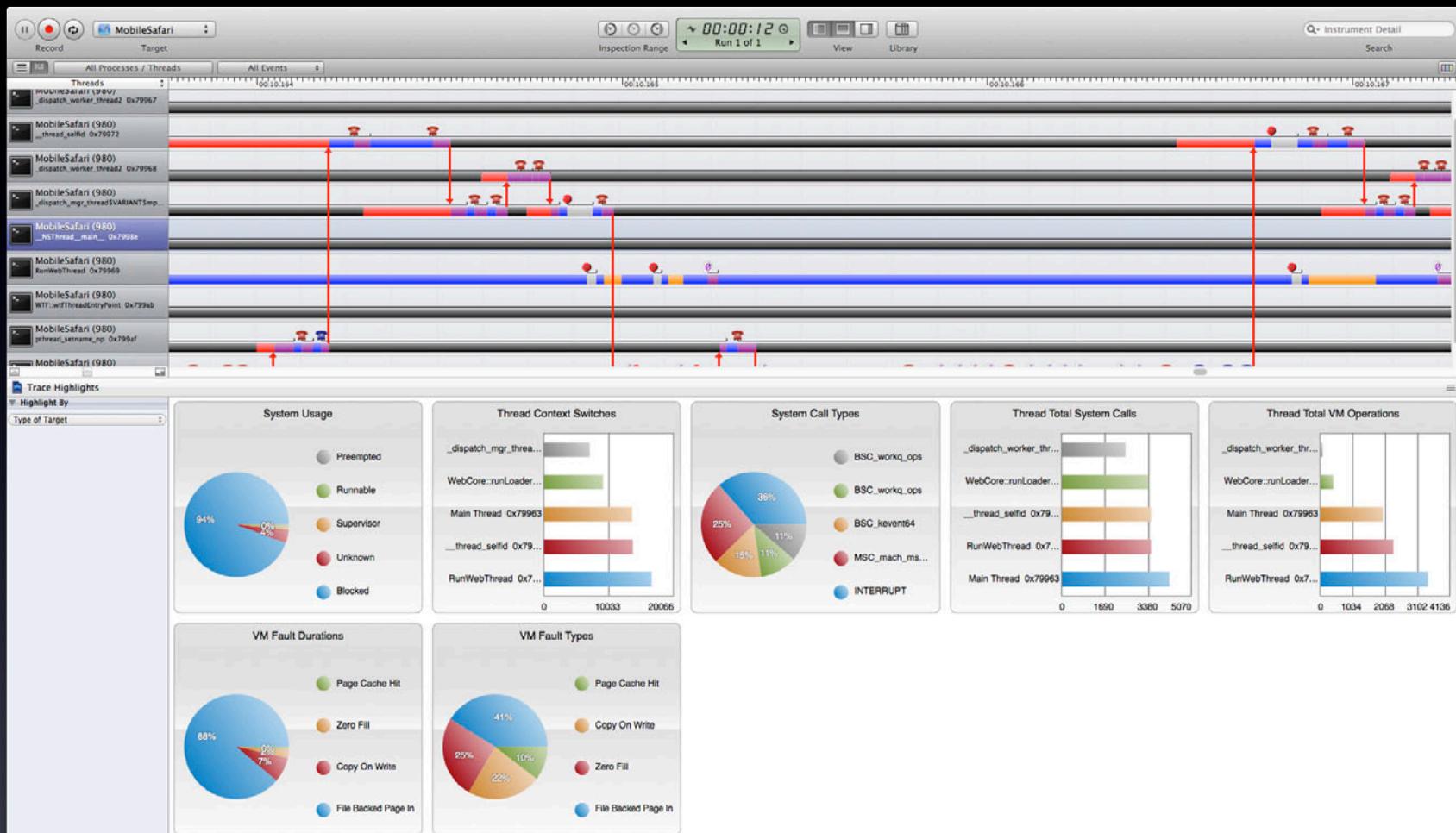
LLDB

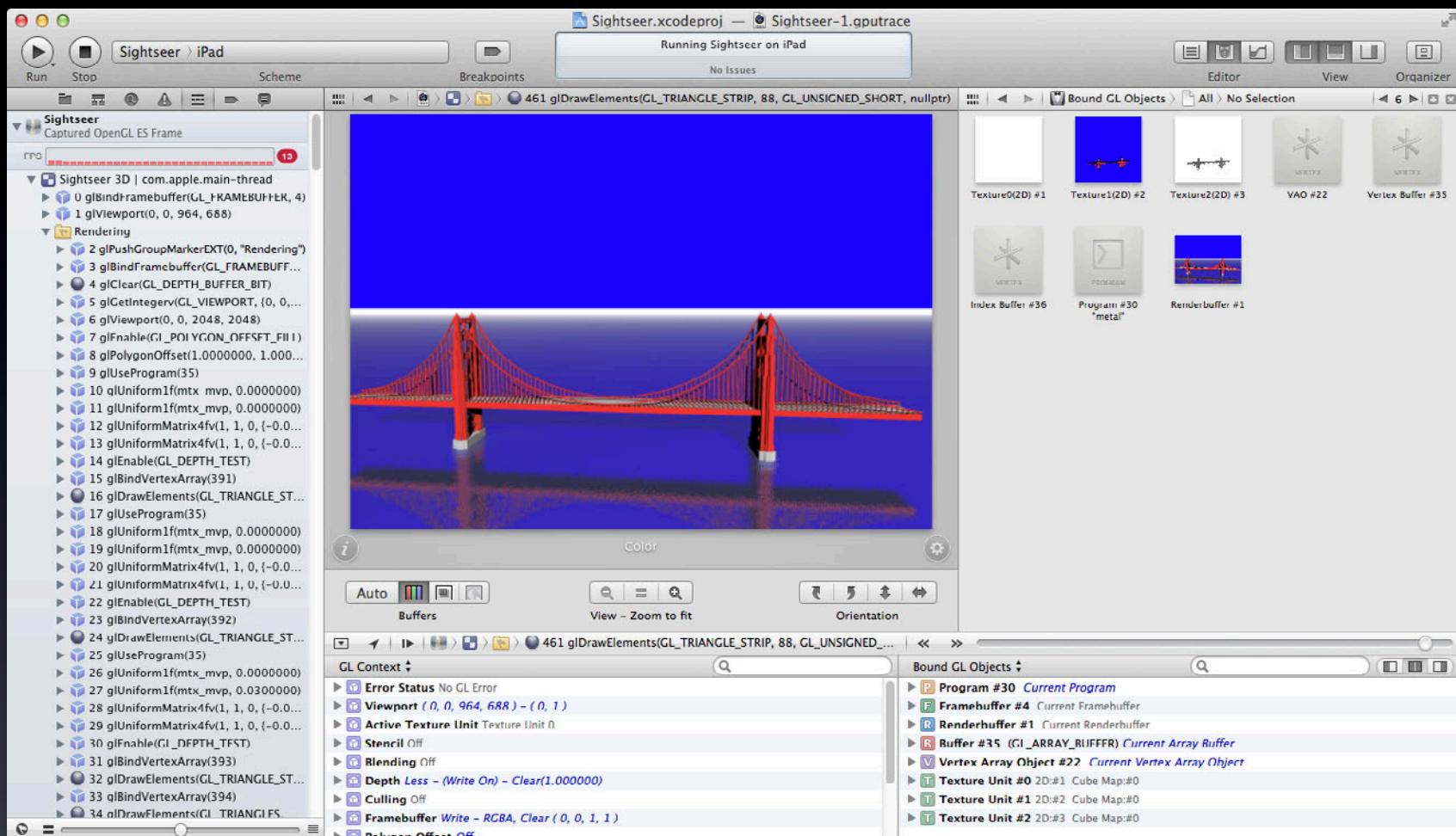


LLDB

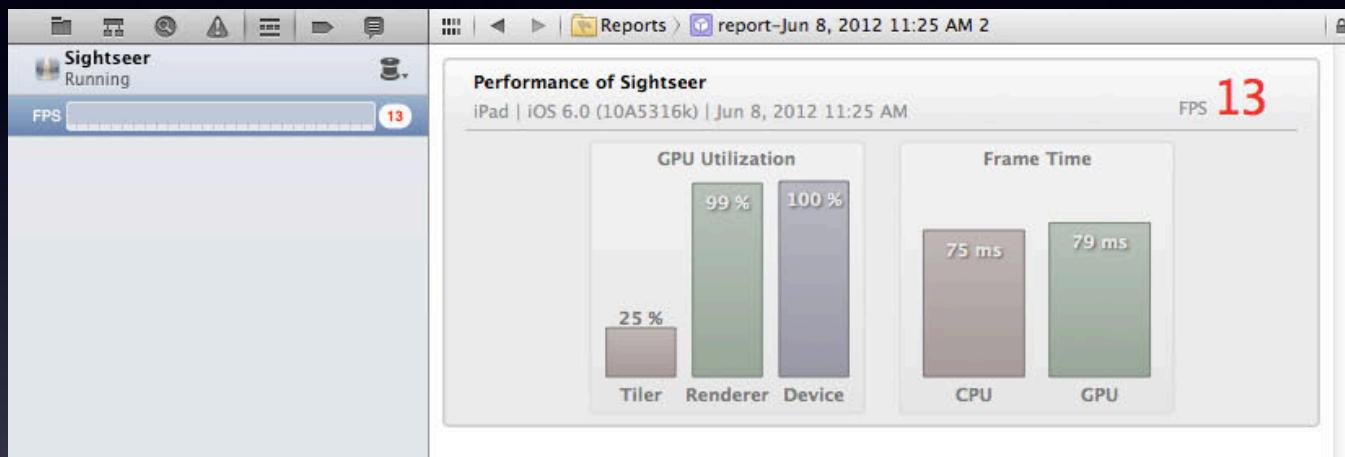


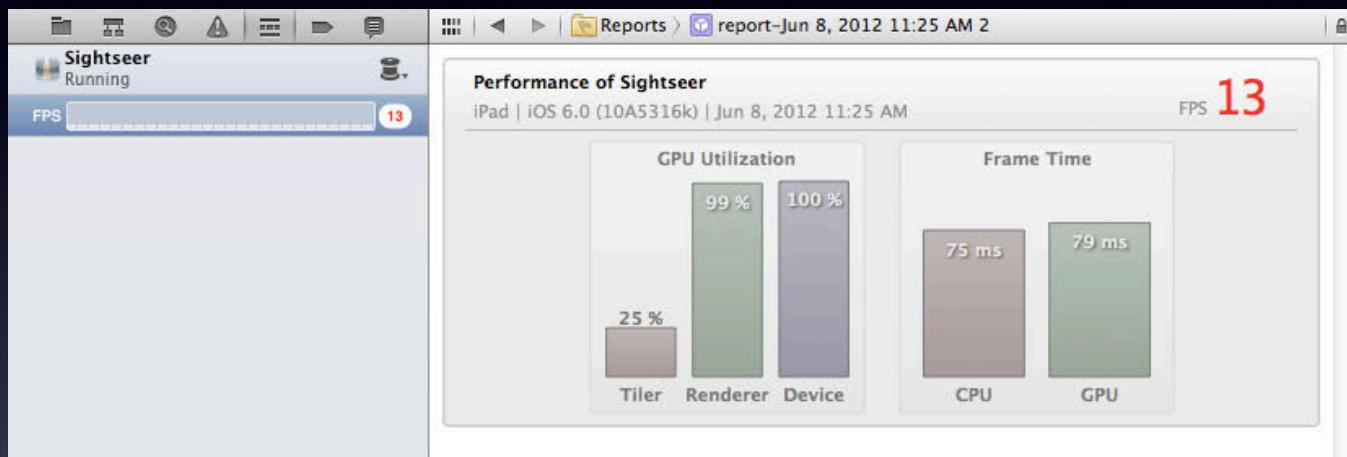
OpenGL|ES











Demonstration

Ted Kremenek
Senior Engineering Manager, LLVM Compiler

```
60  
61  
62  
63 → void foo(int *P, int Condition) { ← 3. Entered call from 'bar'  
64     if (Condition)  
65         *P = 0;  
66 }  
67  
68  
69     int bar() {  
70         int Value; ← 1. Variable 'Value' declared without an initial value  
71         ← 2. Calling 'foo'  
72         foo(&Value, 0); ← 2  
73     }  
74     return Value; ← 5. Undefined or garbage value returned to caller  
75 }  
76  
77  
78
```

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

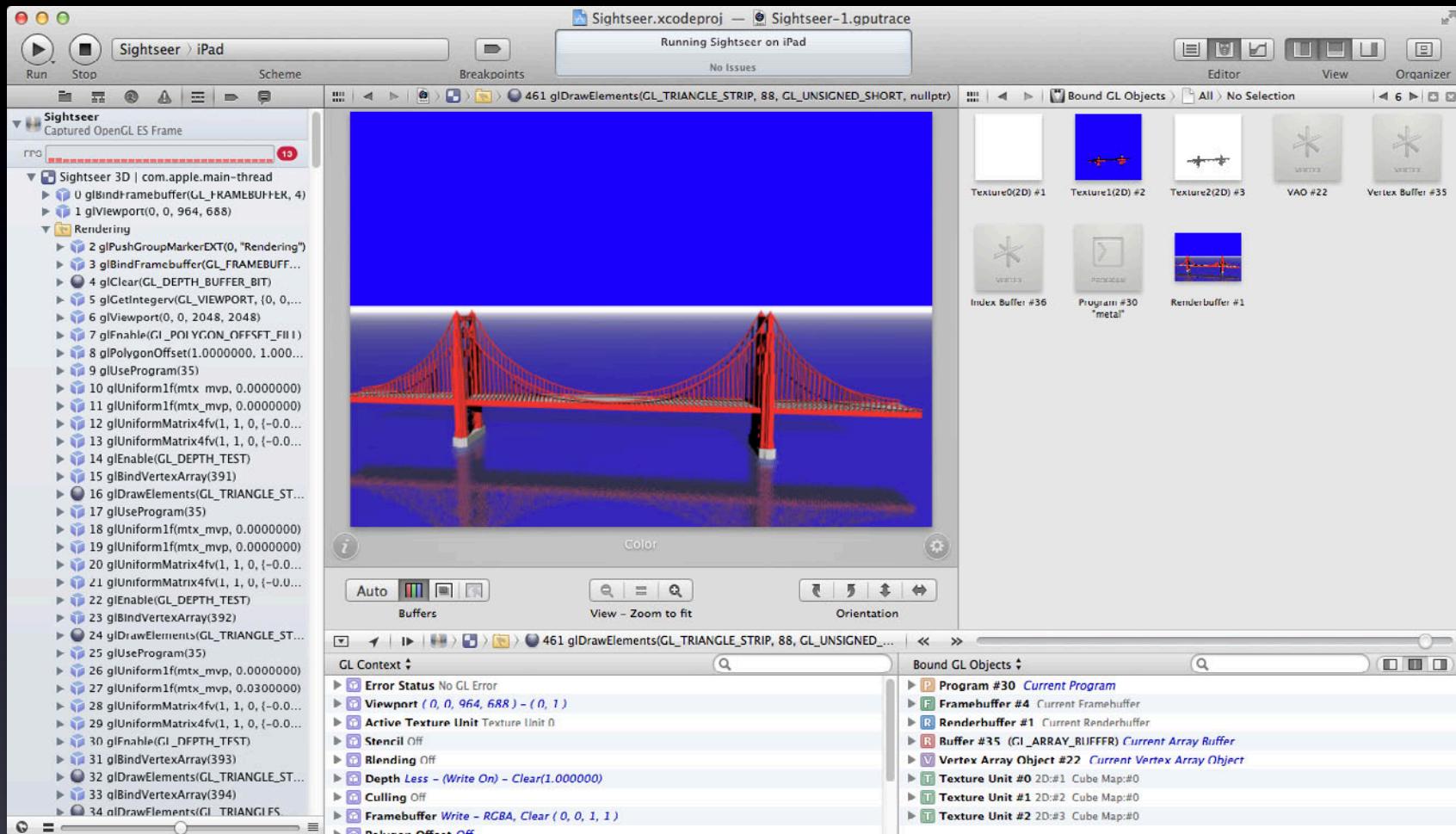
▼ ▶ ⌂ ⌄ ⌅ ⌆ ⌇ CoverageTest > Thread

Auto All Output

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

For more information on any particular command, try
'help <command-name>'.

(lldb) █



Sightseer.xcodeproj

Running Sightseer on iPad

No Issues

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 %	98 %
Renderer	99 %	
Device		

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_TRIANGLE_STRIP, 10, GL_UNSIGNED_SHORT, nullptr)
 - 319 glDrawElements(GL_TRIANGLE_STRIP, 10, GL_UNSIGNED_SHORT, nullptr)

Analyze Performance

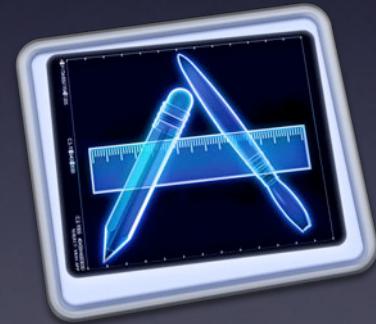
GL Context: No Selection

```

29 varying highp vec4 shadowSpacePos;
30 uniform highp float wavePhase;
31 uniform sampler2DShadow 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 = shadow2DProjFXT(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     {
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

Bound GL Objects:



LLDB



OpenGL|ES

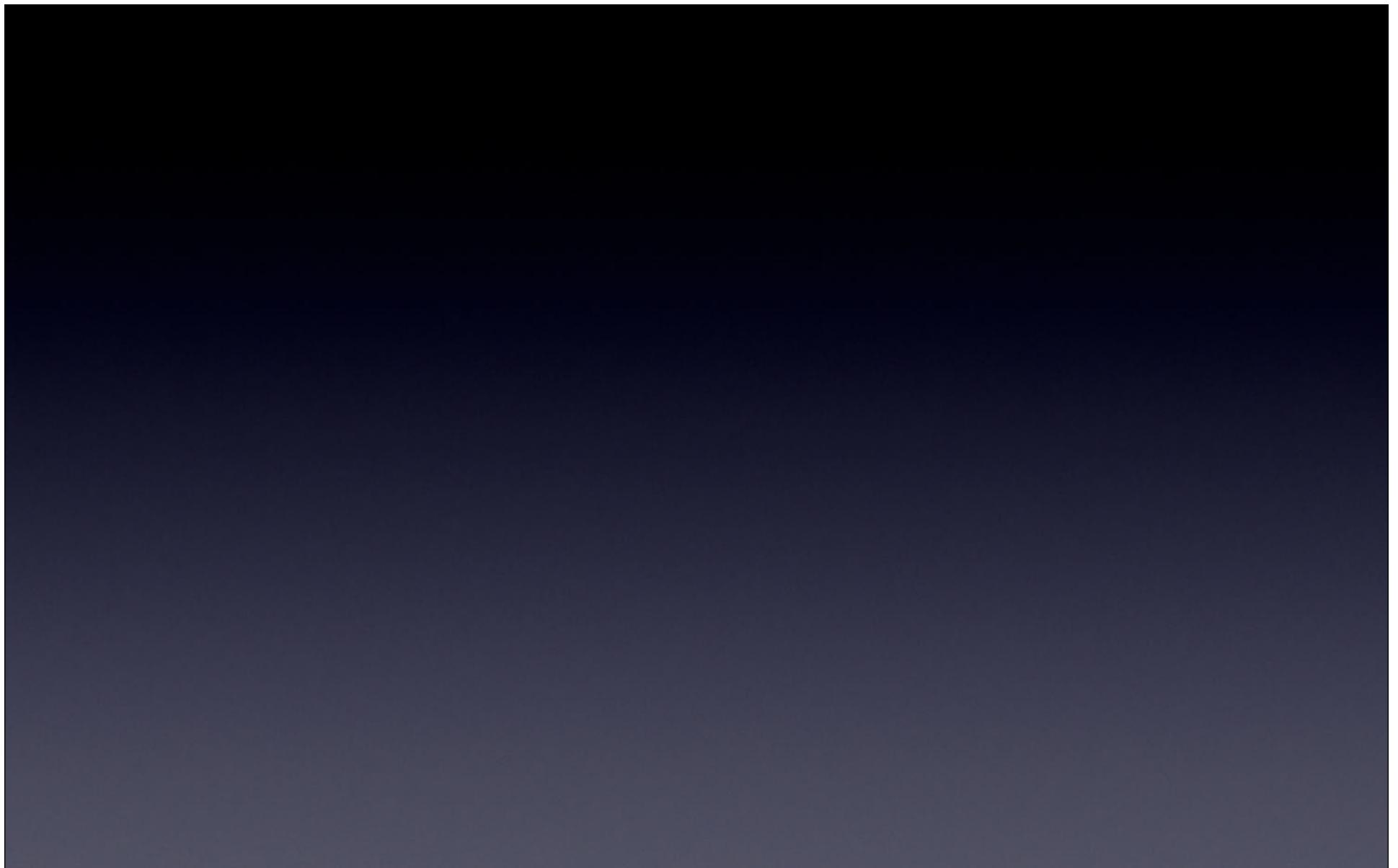


Optim!ze

Deploy

Deploy





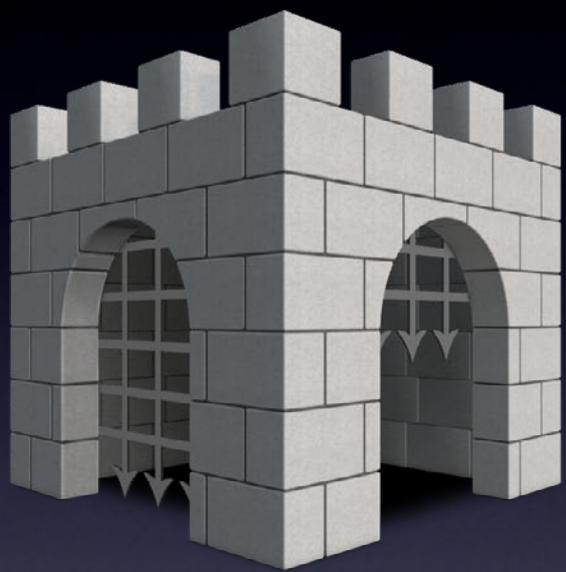


App Store



Developer ID







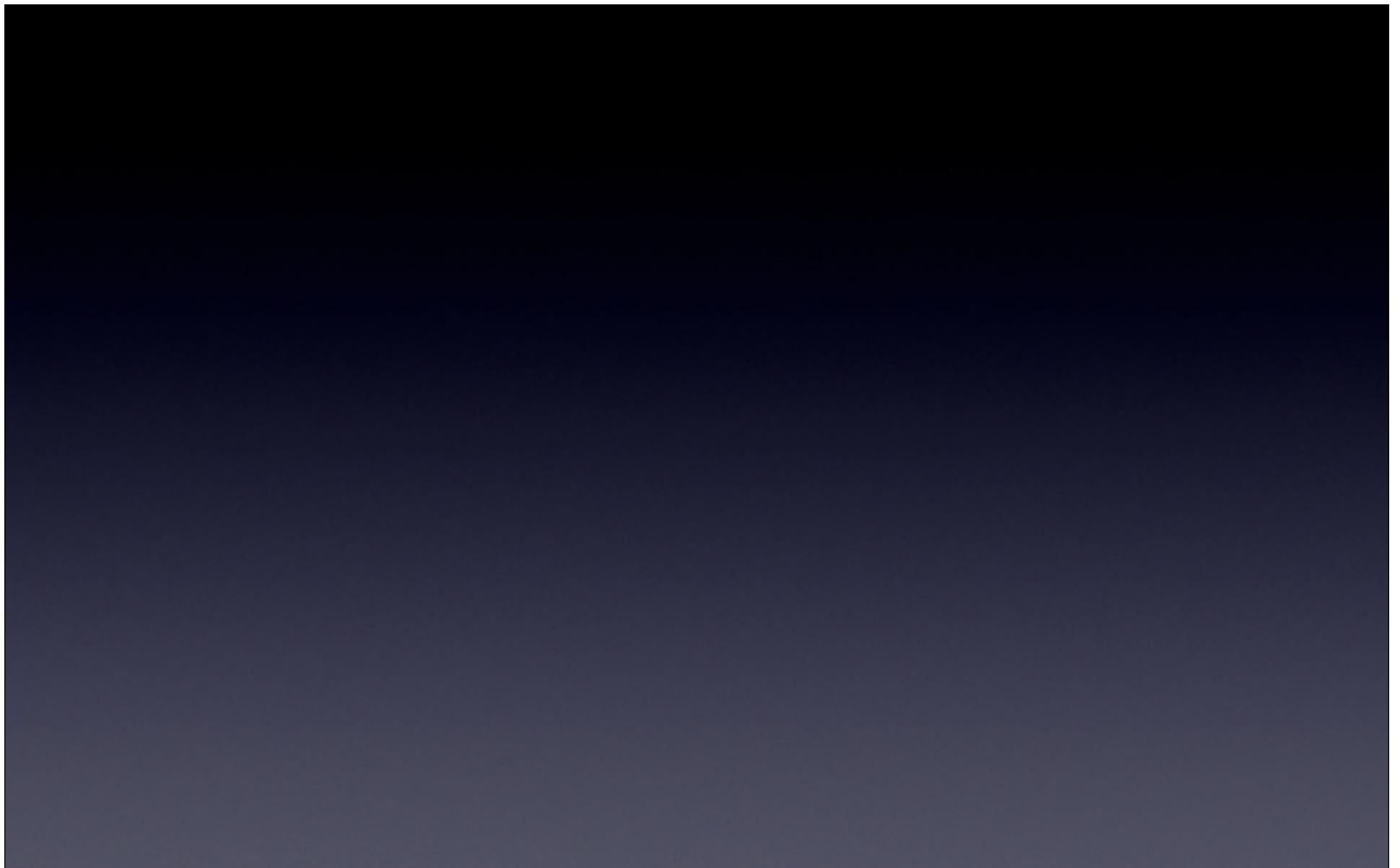
App Store

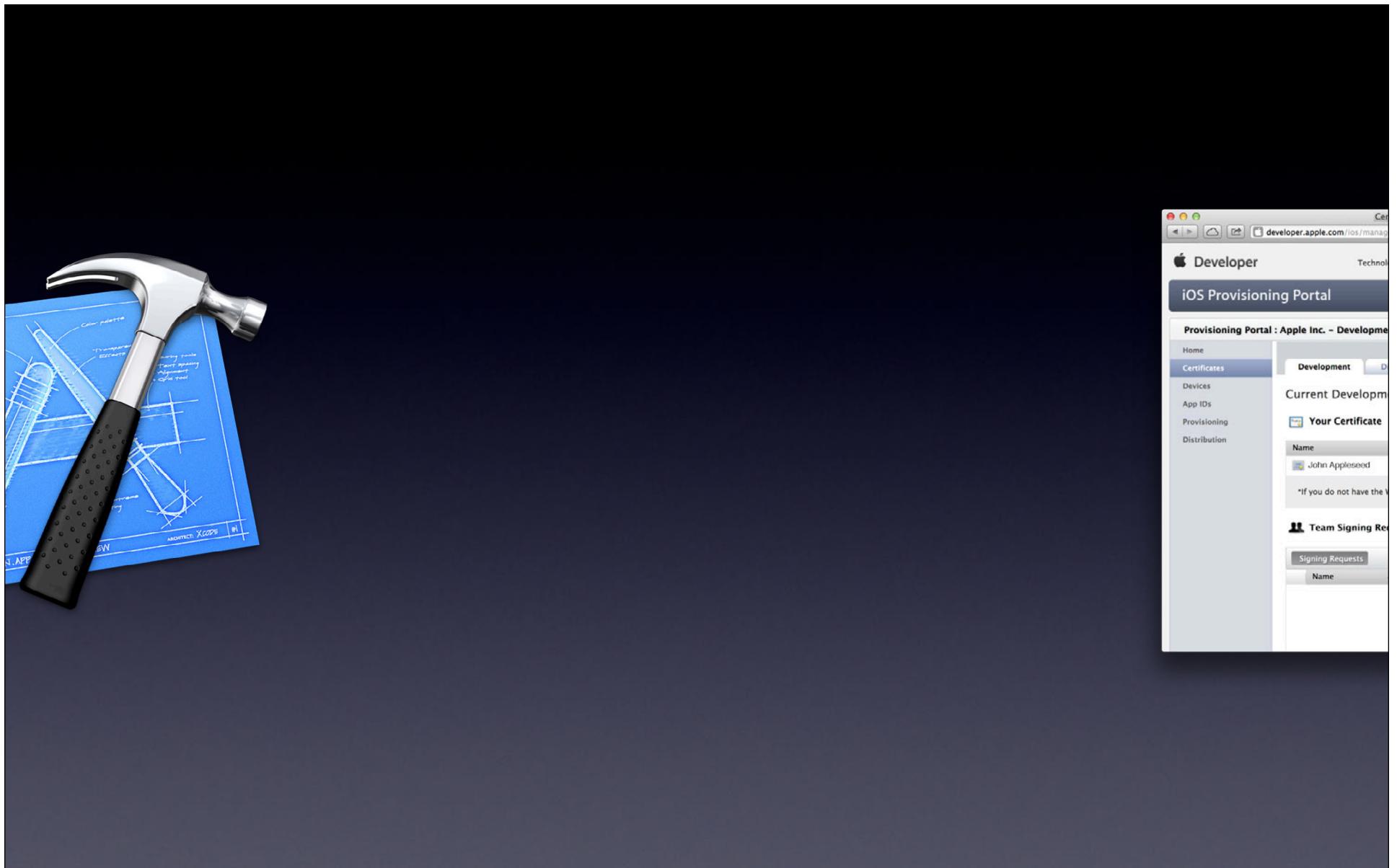


Developer ID



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 — Analyze Succeeded

Sightseer › iOS Device

Sightseer.xcodeproj Debugger SGTLandmarksView

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

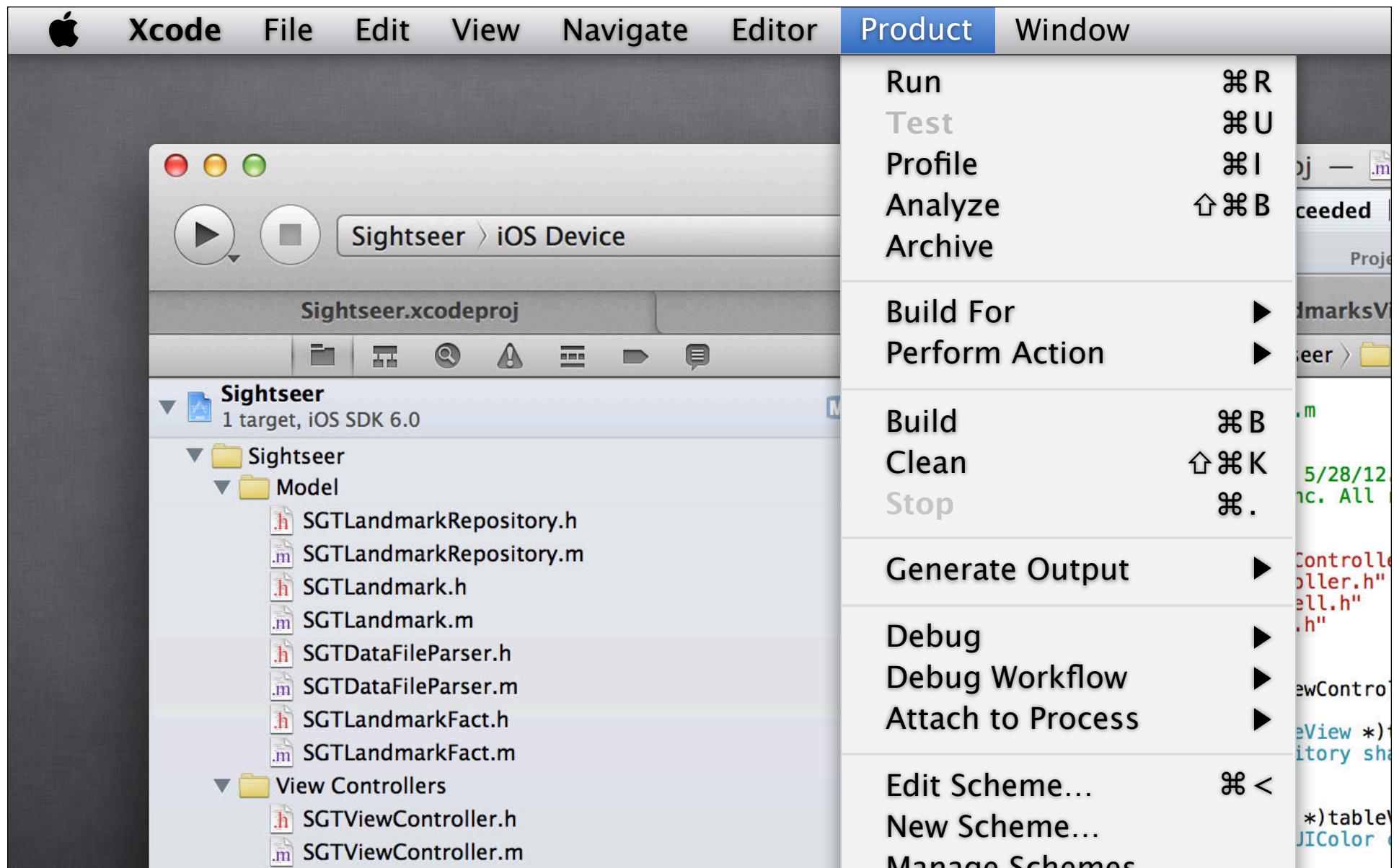
```
// SGTLocationsViewController.m
// Sightseer
//
// Created by Kevin Cathey on 5/28/12
// Copyright (c) 2012 Apple Inc. All rights reserved.

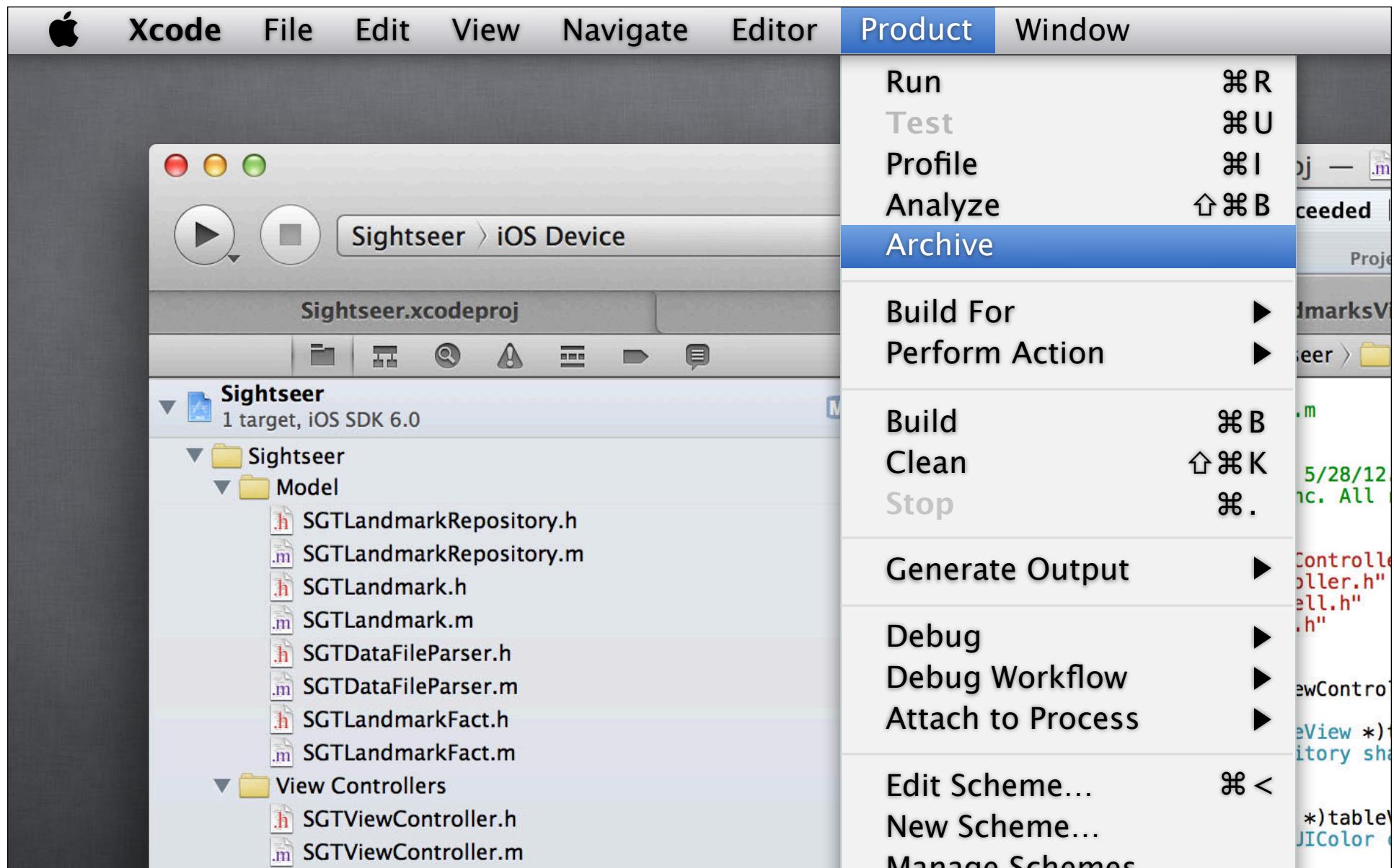
#import "SGTLandmarkDetailViewController.h"
#import "SGTLandmarksViewController.h"
#import "SGTLandmarkTableViewCell.h"
#import "SGTLandmarkRepository.h"
#import "SGTLandmark.h"

@implementation SGTLandmarksViewController

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
    return [[[SGTLandmarkRepository sharedRepository] objects] count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:@"SGTLandmarkCell"];
    if (cell == nil)
    {
        cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:@"SGTLandmarkCell"];
    }
    SGTLandmark *landmark = [[SGTLandmarkRepository sharedRepository] objectAtIndex:indexPath.row];
    cell.textLabel.text = landmark.name;
    cell.detailTextLabel.text = landmark.description;
    return cell;
}
```







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



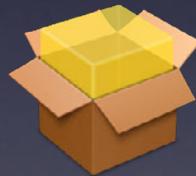
Mac App Store



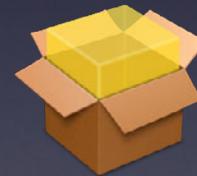
iOS App Store



Developer ID



Ad-Hoc Deployment



Enterprise Deployment



Beta Testing



Mac App Store



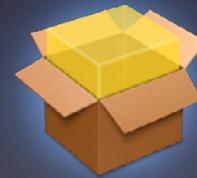
iOS App Store



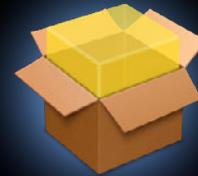
Developer ID



Ad-Hoc Deployment



Enterprise Deployment



Beta Testing

Now





Available Now



Available Now





