

Maximizing Productivity in Xcode 4

Session 306

Mike Ferris
Xcode

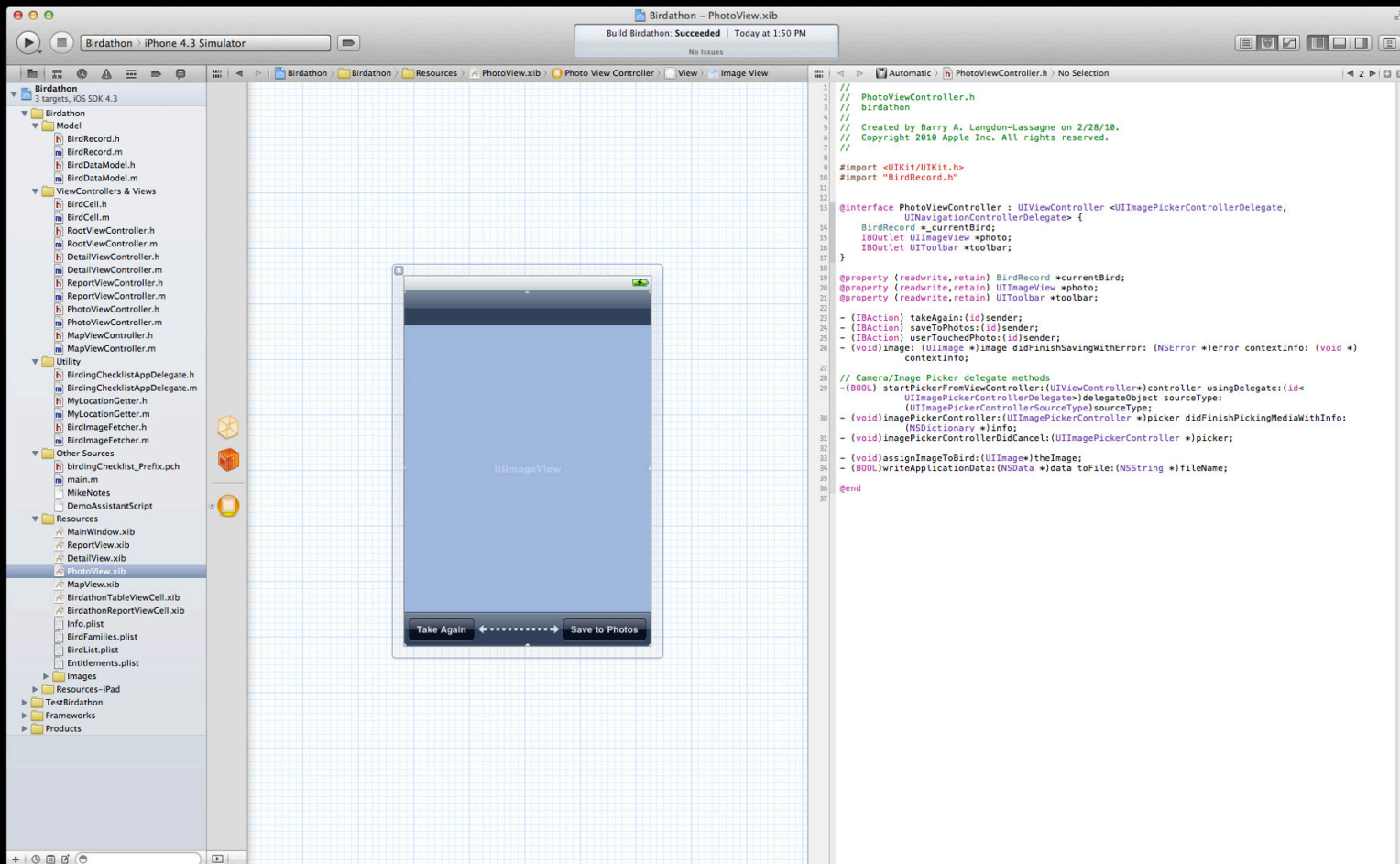
These are confidential sessions—please refrain from streaming, blogging, or taking pictures

Roadmap

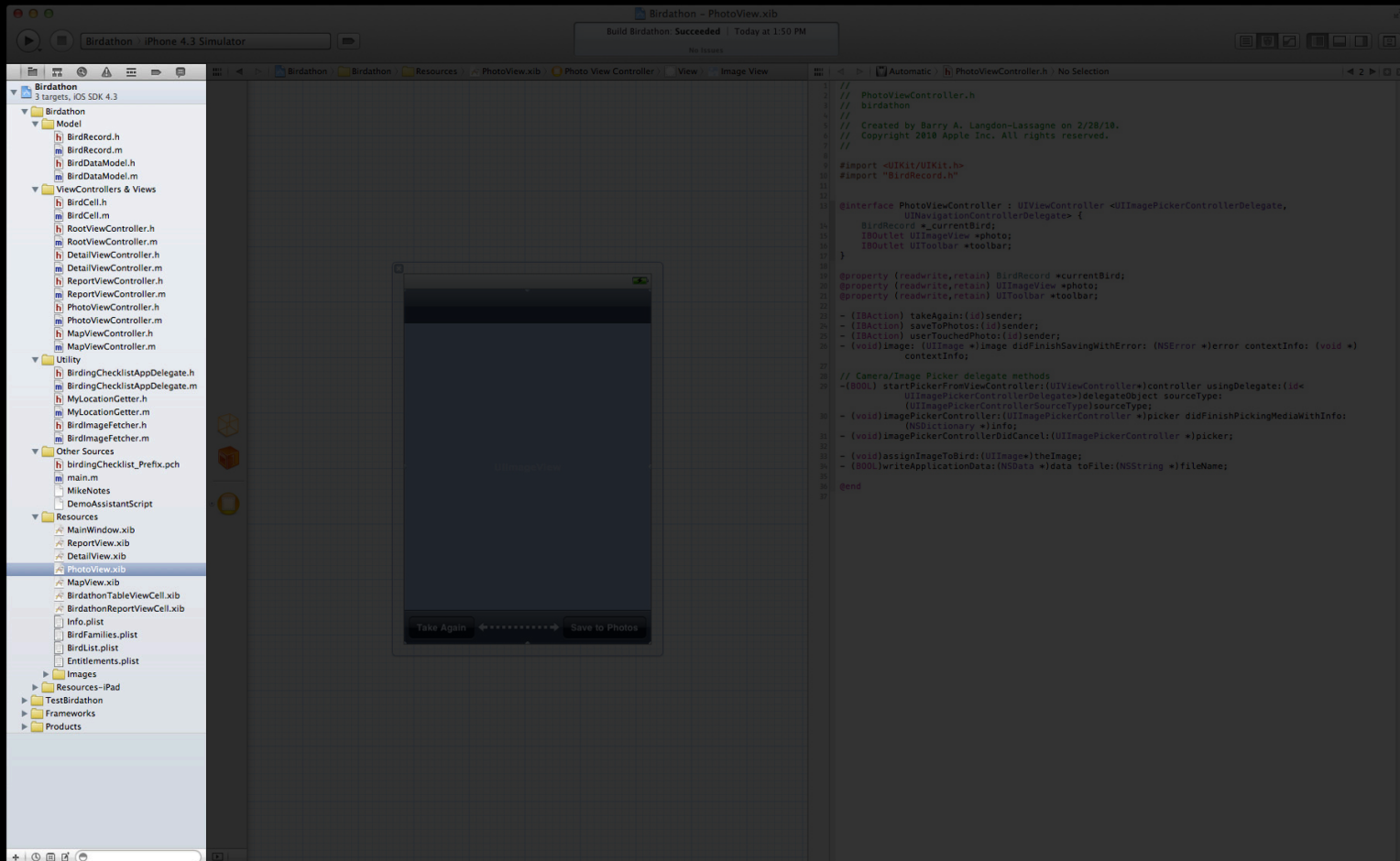
- Working in Xcode
- Customizing Xcode
- Source Editing

Working in Xcode

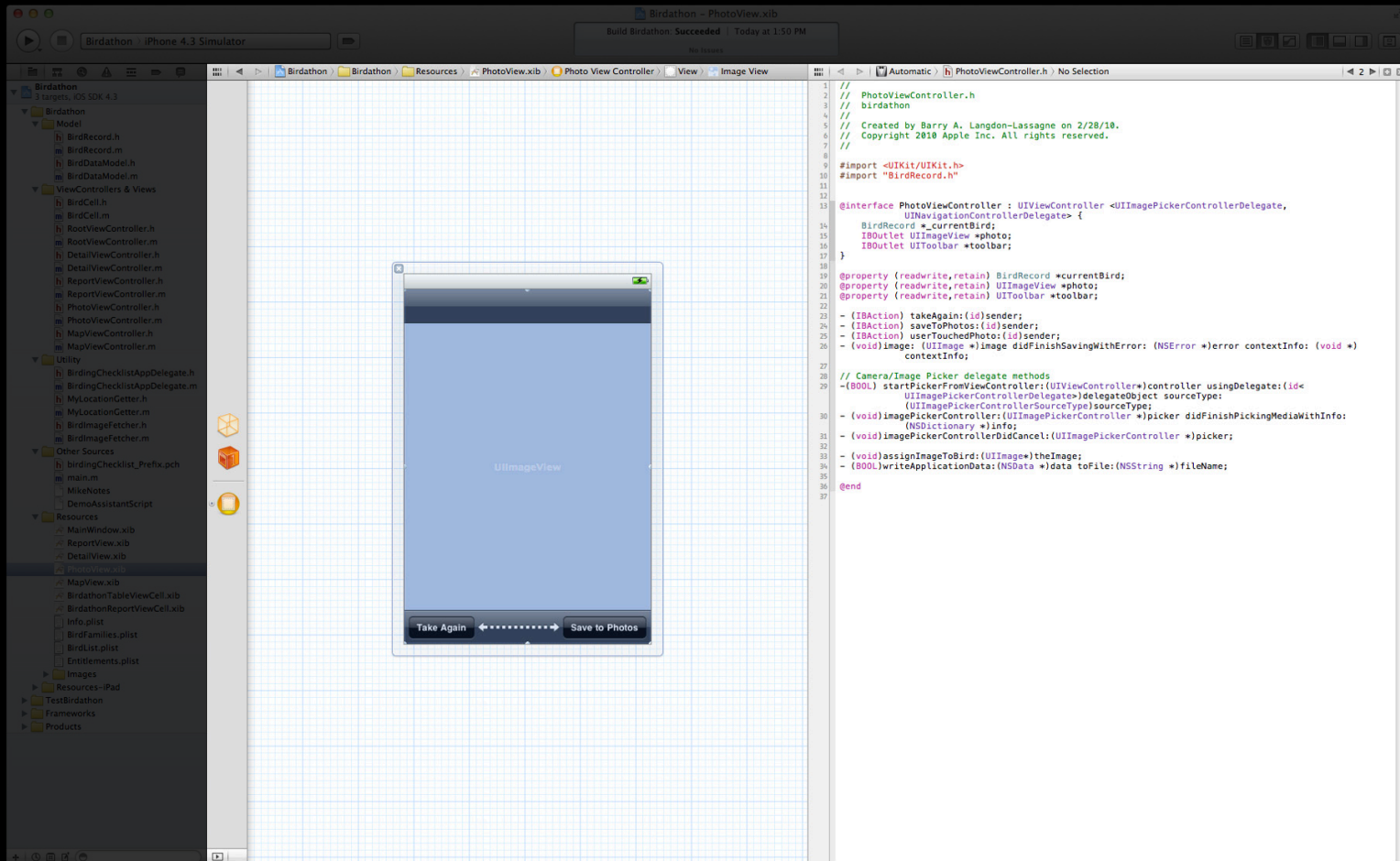
The Xcode Workspace



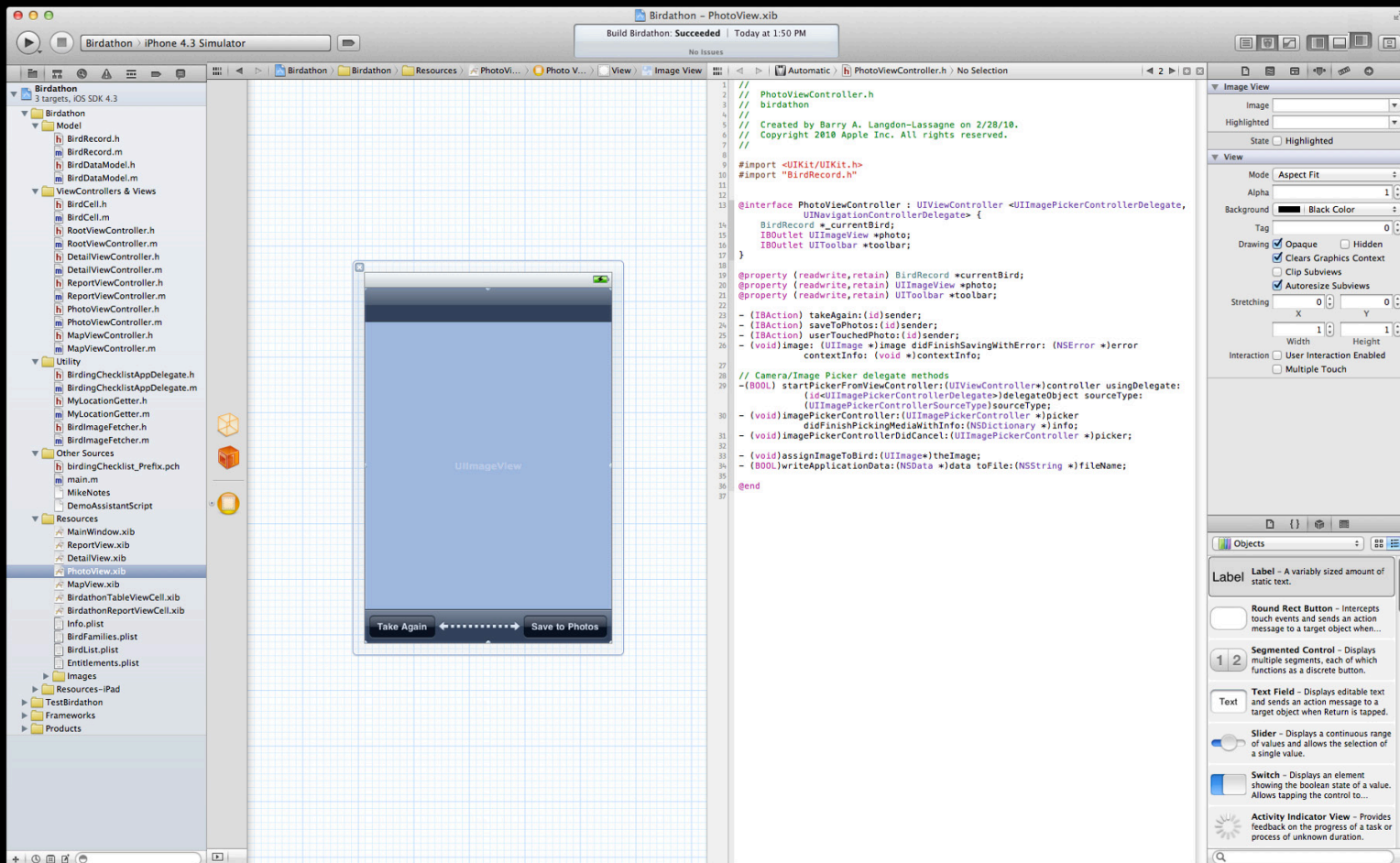
The Xcode Workspace



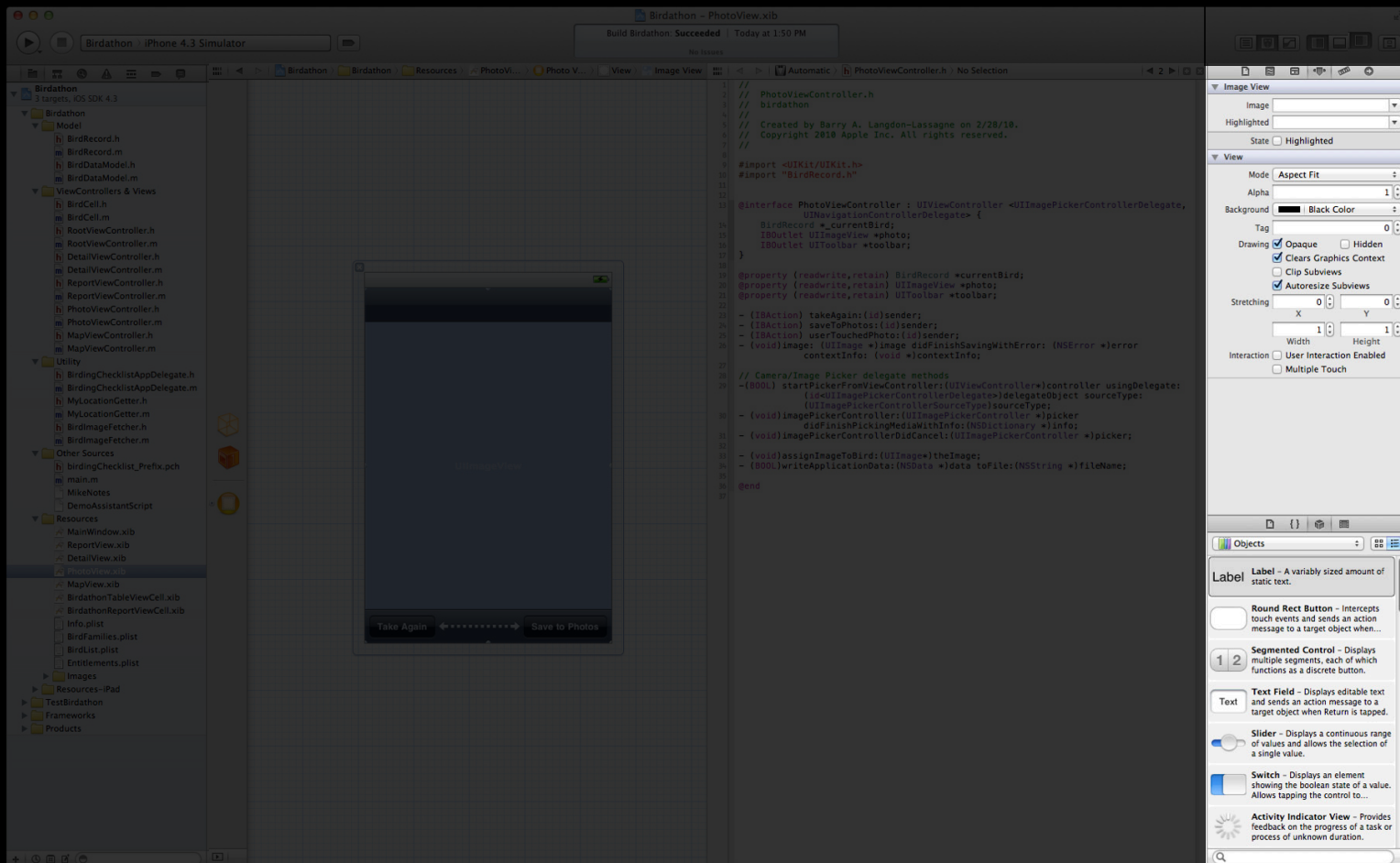
The Xcode Workspace



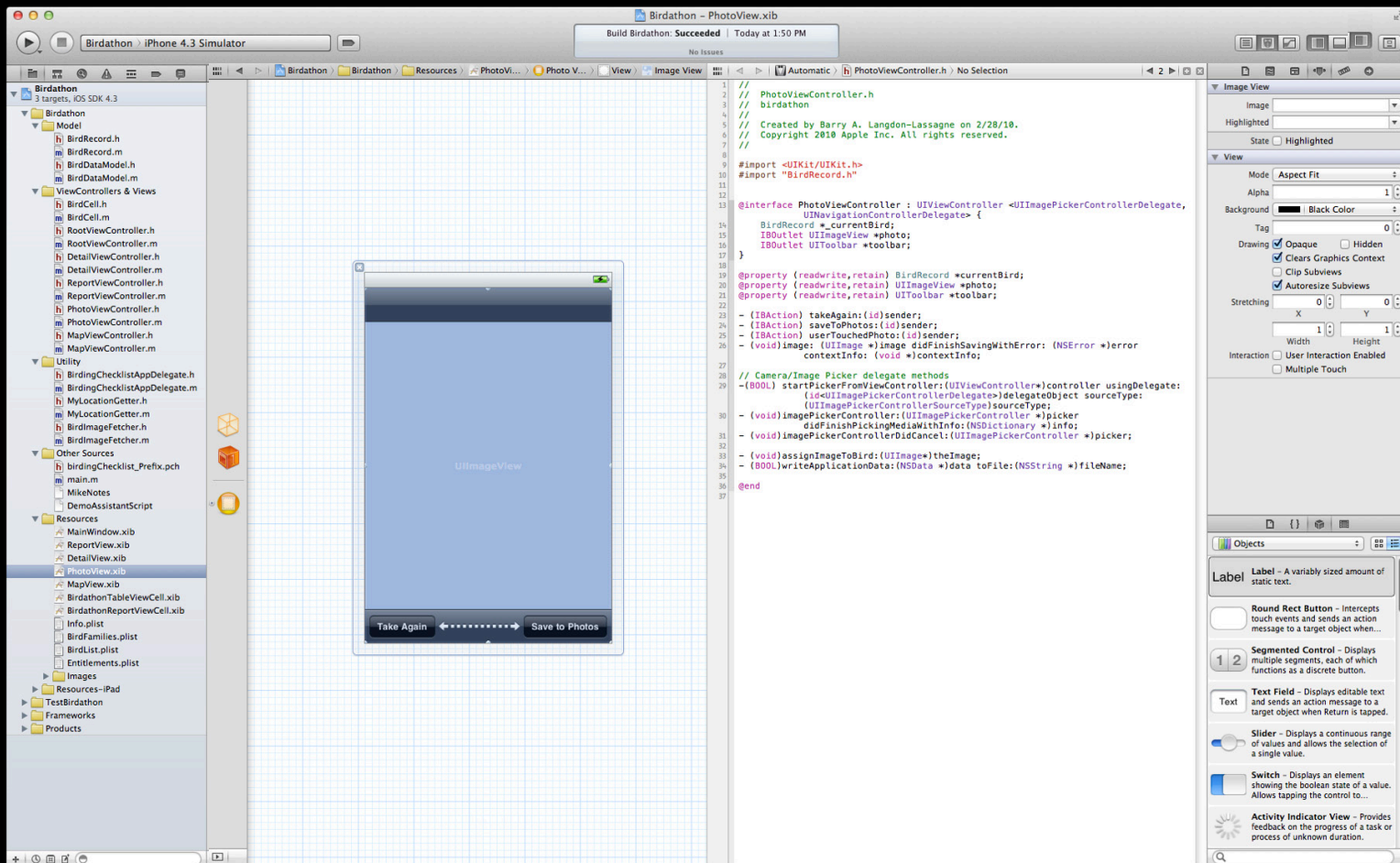
The Xcode Workspace



The Xcode Workspace



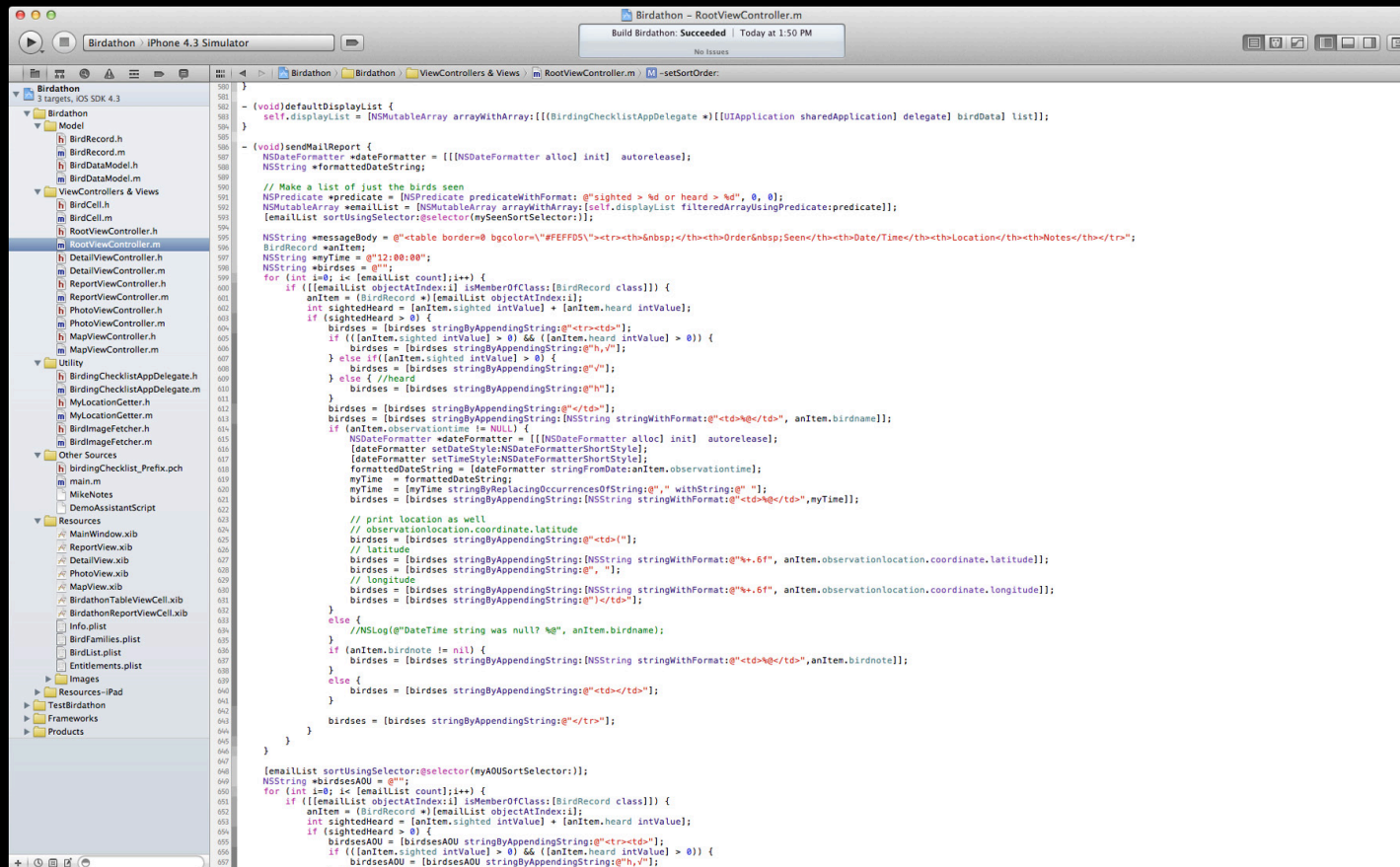
The Xcode Workspace



Working in Xcode

- Progress through activities
- The workspace adapts
- Navigation and actions drive Xcode

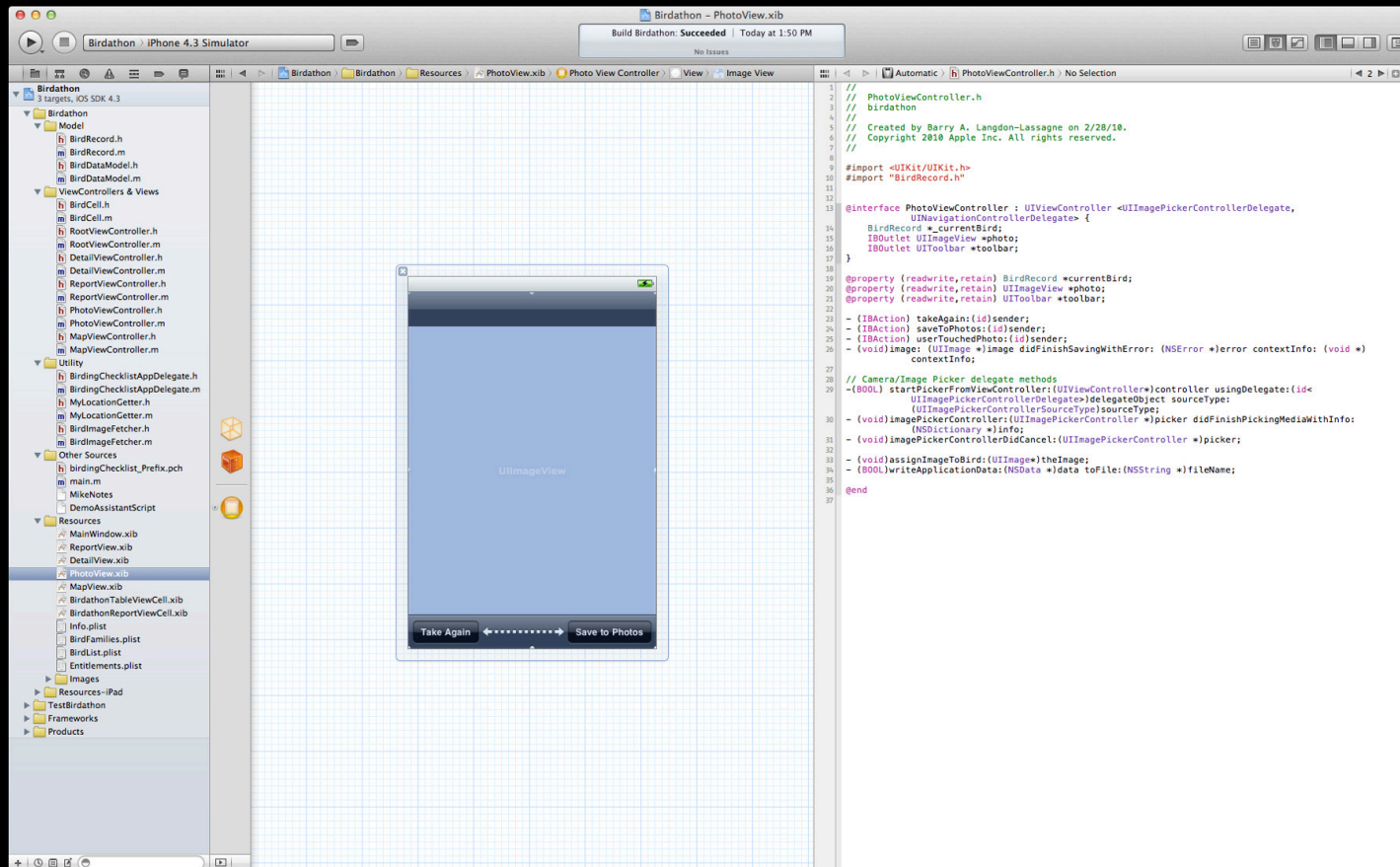
Working in Xcode Browse



```
580 }
581
582 -(void)defaultDisplayList {
583     self.displayList = [NSMutableArray arrayWithArray:[[(BirdingCheckListAppDelegate *)[UIApplication sharedApplication] delegate] birdData] list];
584 }
585
586 -(void)sendMailReport {
587     NSDateFormatter *dateFormatter = [[NSDateFormatter alloc] init] autorelease;
588     NSString *formattedDateString;
589
590     // Make a list of just the birds seen
591     NSPredicate *predicate = [NSPredicate predicateWithFormat:@"sighted > %d or heard > %d", 0, 0];
592     NSMutableArray *emailList = [NSMutableArray arrayWithArray:[self.displayList filteredArrayUsingPredicate:predicate]];
593     [emailList sortUsingSelector:@selector(mySeenSortSelector)];
594
595     NSString *messageBody = @"<table border=0 bgcolor=#FFFFFF><tr><th>&nbsp;</th><th>Order&nbsp;</th><th>Date/Time</th><th>Location</th><th>Notes</th></tr>";
596     BirdRecord *anItem;
597     NSString *myTime = @"12:00:00";
598     NSString *birds = @" ";
599     for (int i=0; i < [emailList count]; i++) {
600         if ([[[emailList objectAtIndex:i] isKindOfClass:[BirdRecord class]]]) {
601             anItem = (BirdRecord *)[emailList objectAtIndex:i];
602             int sightedHeard = [anItem.sighted intValue] + [anItem.heard intValue];
603             if (sightedHeard > 0) {
604                 birds = [birds stringByAppendingString:@"<tr><td>"];
605                 if ([anItem.sighted intValue] > 0) && ([anItem.heard intValue] > 0) {
606                     birds = [birds stringByAppendingString:@"h,v"];
607                 } else if ([anItem.sighted intValue] > 0) {
608                     birds = [birds stringByAppendingString:@"h"];
609                 } else { //heard
610                     birds = [birds stringByAppendingString:@"v"];
611                 }
612                 birds = [birds stringByAppendingString:@"<td>"];
613                 birds = [birds stringByAppendingString:[NSString stringWithFormat:@"<td>%e</td>", anItem.birdname]];
614                 if (anItem.observations != NULL) {
615                     NSDateFormatter *dateFormatter = [[NSDateFormatter alloc] init] autorelease;
616                     [dateFormatter setDateStyle:NSDateFormatterShortStyle];
617                     [dateFormatter setTimeStyle:NSDateFormatterShortStyle];
618                     formattedDateString = [dateFormatter stringFromDate:anItem.observations];
619                     myTime = formattedDateString;
620                     myTime = [myTime stringByReplacingOccurrencesOfString:@" " withString:@""];
621                     birds = [birds stringByAppendingString:[NSString stringWithFormat:@"<td>%e</td>", myTime]];
622
623                     // print location as well
624                     // observationLocation.coordinate.latitude
625                     birds = [birds stringByAppendingString:@"<td>["];
626                     // latitude
627                     birds = [birds stringByAppendingString:[NSString stringWithFormat:@"%+.6f", anItem.observationLocation.coordinate.latitude]];
628                     birds = [birds stringByAppendingString:@" "];
629                     // longitude
630                     birds = [birds stringByAppendingString:[NSString stringWithFormat:@"%+.6f", anItem.observationLocation.coordinate.longitude]];
631                     birds = [birds stringByAppendingString:@"<td>"];
632                 }
633                 else {
634                     //NSLog(@"Date/Time string was null! %e", anItem.birdname);
635                 }
636                 if (anItem.birdnote != nil) {
637                     birds = [birds stringByAppendingString:[NSString stringWithFormat:@"<td>%e</td>", anItem.birdnote]];
638                 }
639                 else {
640                     birds = [birds stringByAppendingString:@"<td></td>"];
641                 }
642                 birds = [birds stringByAppendingString:@"</tr>"];
643             }
644         }
645     }
646
647     [emailList sortUsingSelector:@selector(myAOUSelector)];
648     NSString *birdsAOU = @" ";
649     for (int i=0; i < [emailList count]; i++) {
650         if ([[[emailList objectAtIndex:i] isKindOfClass:[BirdRecord class]]]) {
651             anItem = (BirdRecord *)[emailList objectAtIndex:i];
652             int sightedHeard = [anItem.sighted intValue] + [anItem.heard intValue];
653             if (sightedHeard > 0) {
654                 birdsAOU = [birdsAOU stringByAppendingString:@"<tr><td>"];
655                 if ([anItem.sighted intValue] > 0) && ([anItem.heard intValue] > 0) {
656                     birdsAOU = [birdsAOU stringByAppendingString:@"h,v"];
657                 }
658             }
659         }
660     }
661 }
```

Working in Xcode

Barthoese

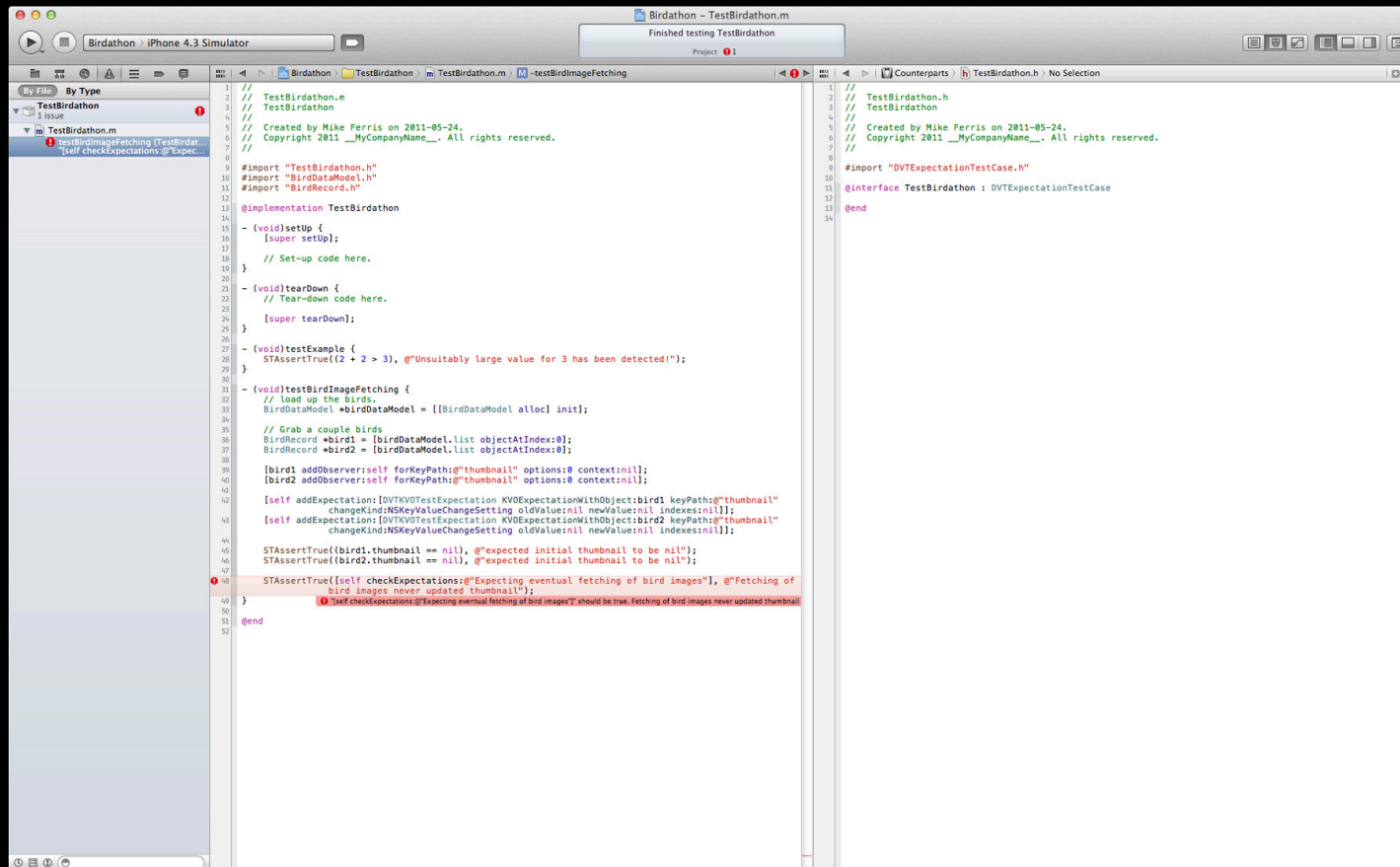


Working in Xcode

Refactor

```
1 // BirdDataModel.m
2 // apex_birdathon
3 // Created by Barry Langdon-Lassagne on 2/1/10.
4 // Copyright 2010 Apple Inc. All rights reserved.
5
6 #import "BirdDataModel.h"
7 #import "BirdRecord.h"
8
9 NSString *DATA_FILENAME = @"BirdList.archive";
10 NSString *INITIAL_DATA_FILENAME = @"BirdList.plist";
11 NSString *BIRD_FAMILIES_FILENAME = @"BirdFamilies.plist";
12 int sortOrder;
13
14 @implementation BirdDataModel
15
16 @synthesize list;
17 @synthesize familiesList;
18 @synthesize origlist;
19
20 - (id) initWithDictionary: {
21     if ([self == [super init]]) {
22         // Look for the data file; if it's not there, create a new array
23         NSString *dataFilePath = [self dataFilePath];
24         sortOrder = 0; // default ADU Order
25         NSDictionary *birdFamiliesDict = [NSDictionary dictionaryWithContentsOfFile:[NSBundle mainBundle]
26             pathForResource:@"BIRD_FAMILIES_FILENAME" ofType:nil];
27         self.familiesList = [birdFamiliesDict objectForKey:@"birdfamilies"]; // get the array
28         of dictionaries (to be BirdRecords) out of the plist dictionary
29         if ([[NSBundle mainBundle] fileExistsAtPath:dataFilePath]) { // second and
30             subsequent launches
31             self.list = [NSKeyedUnarchiver unarchiveObjectWithFile:dataFilePath];
32
33             // sortOrder = [[NSUserDefaults standardUserDefaults] integerForKey:
34                 SORT_ORDER_KEY];
35             // NSLog(@"Sort order key: %d", sortOrder);
36         }
37         else {
38             // Should only be needed for first launch - then should save and restore state
39             [self readBirdDataFromFile];
40         }
41     }
42     return self;
43 }
44
45 - (void) readBirdDataFromFile {
46     NSDictionary *birdDict = [NSDictionary dictionaryWithContentsOfFile:[NSBundle mainBundle]
47         pathForResource:INITIAL_DATA_FILENAME ofType:nil];
48     self.list = [birdDict objectForKey:@"bird"]; // get the array of dictionaries (to be
49         BirdRecords) out of the plist dictionary
50
51     // turn all the dictionary items into BirdRecords
52     for (int i=0; i< [list count]; i++) {
53         [list replaceObjectAtIndex:i withObject:[BirdRecord initWithDictionary:[self
54             list objectAtIndex:i]]];
55     }
56     origlist = [list copy];
57 }
58
59 - (NSString *)dataFilePath {
60     static NSString *dataFilePath = nil;
61     if (dataFilePath == nil) {
62         return dataFilePath;
63     }
64 }
65
66 NSComparisonResult dat = [self.observations compare:otherBird.observations];
```

Working in Xcode Refactor



The screenshot shows the Xcode IDE with a Swift test file named `TestBirdImageFetching`. The code includes imports for `TestBirdathon.h`, `BirdDataModel.h`, and `BirdRecord.h`. It implements `TestBirdathon` with methods for `setUp`, `tearDown`, and a test `testBirdImageFetching`. The test sets up a `BirdDataModel` with two birds, `bird1` and `bird2`, and adds observers for their `thumbnail` properties. It then asserts that `bird1.thumbnail` is `nil` and `bird2.thumbnail` is `nil`. A red error message is visible at the bottom of the test method: `TestBirdImageFetching: [self checkExpectations:@"Expecting eventual fetching of bird images", @"Fetching of bird images never updated thumbnail"] should be true. Fetching of bird images never updated thumbnail`. The Xcode interface also shows a simulator window on the left and a project navigator on the right.

```
1 //
2 // TestBirdathon.m
3 // TestBirdathon
4 //
5 // Created by Mike Ferris on 2011-05-24.
6 // Copyright 2011 __MyCompanyName__. All rights reserved.
7 //
8
9 #import "TestBirdathon.h"
10 #import "BirdDataModel.h"
11 #import "BirdRecord.h"
12
13 @implementation TestBirdathon
14
15 -(void)setup {
16     [super setup];
17     // Set-up code here.
18 }
19
20 -(void)tearDown {
21     // Tear-down code here.
22     [super tearDown];
23 }
24
25
26
27 -(void)testExample {
28     STAssertTrue(2 + 2 > 3, @"Unsuitably large value for 3 has been detected!");
29 }
30
31 -(void)testBirdImageFetching {
32     // load up the birds.
33     BirdDataModel *birdDataModel = [[BirdDataModel alloc] init];
34
35     // Grab a couple birds
36     BirdRecord *bird1 = [birdDataModel.list objectAtIndex:0];
37     BirdRecord *bird2 = [birdDataModel.list objectAtIndex:1];
38
39     [bird1 addObserver:self forKeyPath:@"thumbnail" options:0 context:nil];
40     [bird2 addObserver:self forKeyPath:@"thumbnail" options:0 context:nil];
41
42     [self addExpectation:[DVTKVOExpectation KVOExpectationWithObject:bird1 keyPath:@"thumbnail"
43     changeKind:NSKeyValueChangeSetting oldValue:nil newValue:nil indexes:nil]];
44     [self addExpectation:[DVTKVOExpectation KVOExpectationWithObject:bird2 keyPath:@"thumbnail"
45     changeKind:NSKeyValueChangeSetting oldValue:nil newValue:nil indexes:nil]];
46
47     STAssertTrue(bird1.thumbnail == nil, @"expected initial thumbnail to be nil");
48     STAssertTrue(bird2.thumbnail == nil, @"expected initial thumbnail to be nil");
49
50     STAssertTrue([self checkExpectations:@"Expecting eventual fetching of bird images", @"Fetching of
51     bird images never updated thumbnail"]);
52 }
53
54 @end
```

Working in Xcode

Testing

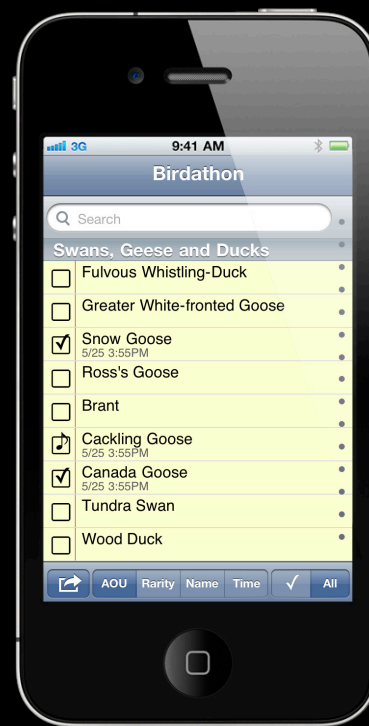
```
1 //
2 // TestBirdation.m
3 // TestBirdation
4 //
5 // Created by Mike Ferris on 2011-05-24.
6 // Copyright 2011 __MyCompanyName__. All rights reserved.
7 //
8
9 #import "TestBirdation.h"
10 #import "BirdDataModel.h"
11 #import "BirdRecord.h"
12
13 @implementation TestBirdation
14
15 -(void)setup {
16     [super setup];
17     // Set-up code here.
18 }
19
20 -(void)tearDown {
21     // Tear-down code here.
22     [super tearDown];
23 }
24
25
26
27 -(void)testExample {
28     STAssertTrue(2 + 2 > 3, @"Unsuitably large value for 3 has been detected!");
29 }
30
31 -(void)testBirdImageFetching {
32     // load up the birds.
33     BirdDataModel *birdDataModel = [[BirdDataModel alloc] init];
34
35     // Grab a couple birds
36     BirdRecord *bird1 = [birdDataModel.list objectAtIndex:0];
37     BirdRecord *bird2 = [birdDataModel.list objectAtIndex:1];
38
39     [bird1 addObserver:self forKeyPath:@"thumbnail" options:0 context:nil];
40     [bird2 addObserver:self forKeyPath:@"thumbnail" options:0 context:nil];
41
42     [self addExpectation:[DVTKVOExpectation KVOExpectationWithObject:bird1 keyPath:@"thumbnail"
43     changeKind:NSKeyValueChangeSetting oldValue:nil newValue:nil indexes:nil]];
44     [self addExpectation:[DVTKVOExpectation KVOExpectationWithObject:bird2 keyPath:@"thumbnail"
45     changeKind:NSKeyValueChangeSetting oldValue:nil newValue:nil indexes:nil]];
46
47     STAssertTrue([bird1.thumbnail == nil], @"expected initial thumbnail to be nil");
48     STAssertTrue([bird2.thumbnail == nil], @"expected initial thumbnail to be nil");
49
50     STAssertTrue([self checkExpectations:@"Expecting eventual fetching of bird images"], @"Fetching of
51     bird images never updated thumbnail");
52     [self checkExpectations:@"Expecting eventual fetching of bird images" should be true. Fetching of bird images never updated thumbnail"];
53 }
54
55 @end
```

Working in Xcode

- Progress through activities
- The workspace adapts
- Navigation and actions drive Xcode

Birdathon

Meet our demo app



Demo

Working in Xcode

How Xcode Works

- Many disparate activities
- Tons of different content
- “Bring it here!”
- Flow through your work
- Navigation brings content to you

Nav·i·gate |'navi,gāt|

1. [no obj.] chart and direct the course of a ship, esp. by using instruments or maps
2. [no obj.] chart a course through the content of a workspace to achieve a particular goal

Customizing Xcode

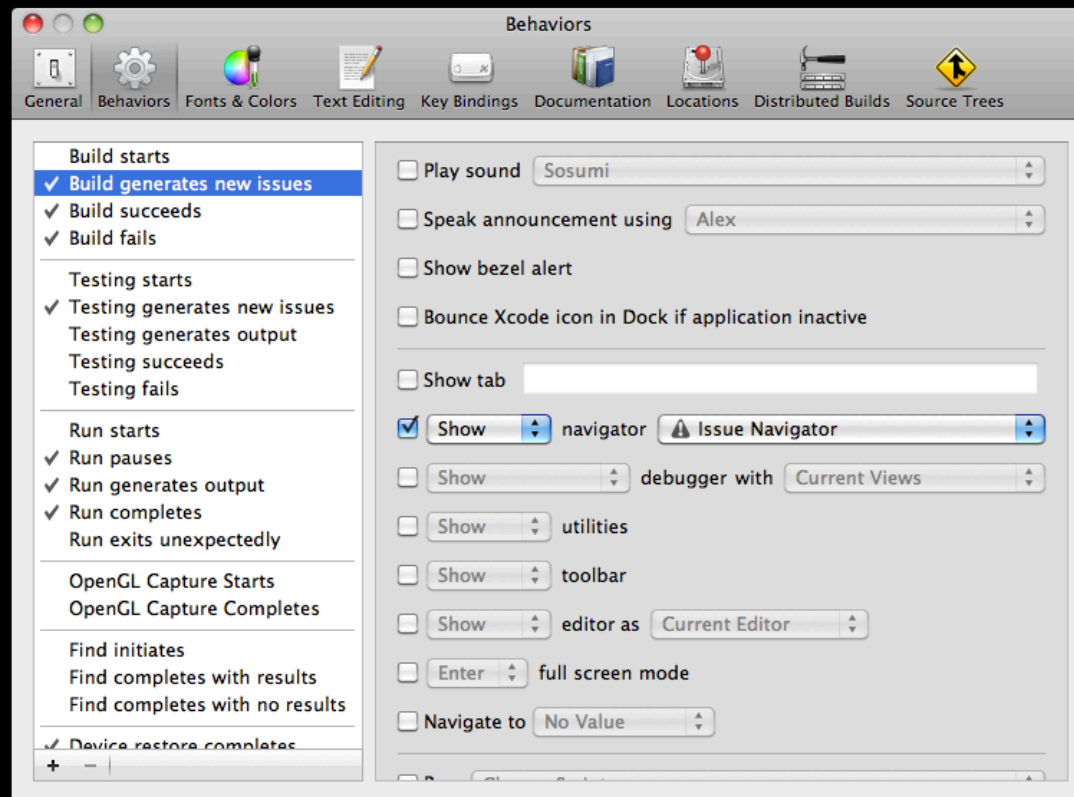
Brooke Callahan
Xcode

Customizing Xcode

- Configure Behaviors for your workflow
- Change how navigation works

What Are Behaviors?

Behaviors are how Xcode reacts to events



Configuring Behaviors

- Run alerts
- Reconfigure UI
- Create a new tab
- Run scripts
- Add new Behaviors for keyboard shortcuts

Demo

Customizing Behaviors

What Is Modified Navigation?

- Navigate in receiving editor or primary editor by default
- Modified Navigation
 - Option modified for pre-configured destination
 - Option+Shift to choose
 - Double-click for new tab/window

Where Does Modified Navigation Work?

The screenshot displays the Xcode IDE with a Swift file open. A 'Navigate' menu is open on the left, listing various navigation actions. Several actions are highlighted with yellow boxes: 'Go Forward', 'Go Back', 'Jump to Selection', 'Jump to Definition', 'Jump to Next Counterpart', 'Jump to Previous Counterpart', 'Jump to Next Placeholder', and 'Jump to Previous Placeholder'. The main editor shows a Swift file with a breakpoint at line 71. A tooltip for 'NSNotificationCenter' is visible, showing its name, availability, abstract, and declared in. An 'Open Quickly' dialog is also open, showing a list of files to open, with 'Foundation/NSString.h' selected. The bottom of the screen shows the debugger console with the output of the current step.

Navigate

- Reveal in Project Navigator ⌘⇧J
- Reveal in Symbol Navigator ⌘⇧⌘
- Reveal in Debug Navigator ⌘⇧⌘
- Open in Assistant Editor ⌘⇧⌘
- Open in... ⌘⇧⌘
- Move Focus To Next Area ⌘⇧⌘
- Move Focus To Previous Area ⌘⇧⌘
- Move Focus To Editor... ⌘⇧⌘
- Go Forward ⌘⇧⌘
- Go Back ⌘⇧⌘
- Jump to Selection ⌘⇧⌘
- Jump to Definition ⌘⇧⌘
- Jump to Next Issue ⌘⇧⌘
- Jump to Previous Issue ⌘⇧⌘
- Jump to Next Counterpart ⌘⇧⌘
- Jump to Previous Counterpart ⌘⇧⌘
- Jump in "SKTDocument.m"... ⌘⇧⌘
- Jump to Next Placeholder ⌘⇧⌘
- Jump to Previous Placeholder ⌘⇧⌘

NSNotificationCenter

Name: NSNotificationCenter
Availability: Mac OS X (10.0 and later)
Abstract: An NSNotificationCenter object (or simply, notification center) provides a mechanism for broadcasting information within a program. An NSNotificationCenter object is essentially a notification dispatch table.
Declared In: NSNotificationCenter.h
Reference: NSNotificationCenter Class Reference
Related Documents: Notification Programming Topics

Open Quickly

- Foundation/NSString.h
- Foundation/NSString.h
- AppKit/NSStringDrawing.h
- AppKit/NSStringDrawing.h
- binutils/dyn-string.h
- binutils/dyn-string.h
- Kernel/libkern/c++/OSString.h
- Kernel/libkern/c++/OSString.h
- AppKit/NSAttributedString.h
- AppKit/NSAttributedString.h
- subversion-1/svn_string.h
- subversion-1/svn_string.h
- Foundation/NSAttributedString.h
- Foundation/NSAttributedString.h
- 10C540
- System/Library/Frameworks/Headers/NSString.h

Debugger Console

Local:

- self = (SKTDocument *) 0x102aaaa60
- _cmd = (SEL) 0x7fff84a0d218 init

All Output:

Copyright © 2004 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB. Type "show warranty" for details.
This GDB was configured as "x86_64-apple-darwin".tty /dev/ttys003
sharedlibrary apply-load-rules all
[Switching to process 30202 thread 0x0]
Single stepping until exit from function __spin_lock, which has no line number information.
(gdb)

Demo

Modified Navigation

Customizing Xcode

- Change Xcode Behaviors to fit your workflow
- Easy to modify how navigation works

On Source Editing

Ron Lue-Sang
Xcode

On Source Editing

- Standard Editor
- Assistant Editor

```
1 /*
2  Document.m
3  Copyright (c) 1995-2009 by Apple Computer, Inc., all rights reserved.
4  Author: David Remahl
5
6  NSDocumentController subclass for TextEdit
7  Required to support transient documents and customized Open panel
8 */
9 /*
10 IMPORTANT: This Apple software is supplied to you by Apple Computer, Inc. ("Apple") in
11 consideration of your agreement to abide by the following terms, and your use, installation,
12 modification or redistribution of this Apple software constitutes acceptance of these
13 terms. If you do not agree with these terms, please do not use, install, modify or
14 redistribute this Apple software.
15
16 In consideration of your agreement to abide by the following terms, and subject to these
17 terms, Apple grants you a personal, non-exclusive license, under Apple's copyrights in
18 this original Apple software (the "Apple Software"), to use, reproduce, modify and
19 redistribute the Apple Software, with or without modifications, in source and/or binary
20 forms; provided that if you redistribute the Apple Software in its entirety and without
21 modifications, you must retain this notice and the following text and disclaimers in all
22 such redistributions of the Apple Software. Neither the name, trademarks, service marks
23 or logos of Apple Computer, Inc. may be used to endorse or promote products derived from
24 the Apple Software without specific prior written permission from Apple. Except as
25 expressly
26 stated in this notice, no other rights or licenses, express or implied, are granted by
27 Apple
28 herein, including but not limited to any patent rights that may be infringed by your
29 derivative works or by other works in which the Apple Software may be incorporated.
30
31 The Apple Software is provided by Apple on an "AS IS" basis. APPLE MAKES NO WARRANTIES,
32 EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF NON-
33 INFRAINGEMENT,
34 MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, REGARDING THE APPLE SOFTWARE OR ITS
35 USE AND OPERATION ALONE OR IN COMBINATION WITH YOUR PRODUCTS.
36
37 IN NO EVENT SHALL APPLE BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL OR CONSEQUENTIAL
38 DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
39 OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) ARISING IN ANY WAY OUT OF THE USE,
40 REPRODUCTION, MODIFICATION AND/OR DISTRIBUTION OF THE APPLE SOFTWARE, HOWEVER CAUSED AND
41 WHETHER UNDER THEORY OF CONTRACT, TORT (INCLUDING NEGLIGENCE), STRICT LIABILITY OR
42 OTHERWISE, EVEN IF APPLE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
43 */
44 #import "DocumentController.h"
45 #import "Document.h"
46 #import "EncodingManager.h"
47 #import "TextEditDefaultsKeys.h"
48 #import "TextEditErrors.h"
49
50 /* A very simple container class which is used to collect the outlets from loading the
51 encoding accessory. No implementation provided, because all of the references are weak
52 and don't need retain/release. Would be nice to be able to switch to a mutable
53 dictionary here at some point.
54 */
55 @interface OpenSaveAccessoryOwner : NSObject {
56
57 }
58 @end
59
60 #import <Cocoa/Cocoa.h>
61 #import "Document.h"
62
63 /* An instance of this subclass is created in the main nib file. */
64
65 // NSDocumentController is subclassed to provide for modification of the open panel.
66 // Normally, there is no need to subclass the document controller.
67 @interface DocumentController : NSDocumentController {
68     NSMutableDictionary *customOpenSettings; // Mapping of document URLs to
69     // encoding, ignore HTML, and ignore rich text settings that override the defaults
70     // from Preferences
71     NSMutableArray *deferredDocuments;
72     NSLock *transientDocumentLock;
73     NSLock *displayDocumentLock;
74 }
75
76 - (NSView *)encodingAccessory:(NSUInteger)encoding includeDefaultEntry:(BOOL)
77 includeDefaultItem encodingPopUp:(NSPopUpButton **)popup checkBox:(NSButton **)button;
78
79 - (Document *)openDocumentWithContentsOfPasteboard:(NSPasteboard *)pb display:(BOOL)
80 display error:(NSError **)error;
81
82 - (NSStringEncoding)lastSelectedEncodingForURL:(NSURL *)url;
83 - (BOOL)lastSelectedIgnoreHTMLForURL:(NSURL *)url;
84 - (BOOL)lastSelectedIgnoreRichForURL:(NSURL *)url;
85
86 - (NSInteger)runModalOpenPanel:(NSOpenPanel *)openPanel forTypes:(NSArray *)types;
87
88 - (Document *)transientDocumentToReplace;
89 - (void)displayDocument:(NSDocument *)doc;
90 - (void)replaceTransientDocument:(NSArray *)documents;
91
92 @end
```

Demo

Standard and assistant editors

More Information

Michael Jurewitz

Development Tools Evangelist
jurewitz@apple.com

Documentation

Xcode 4 User Guide

<http://developer.apple.com/library/ios/#documentation/ToolsLanguages/Conceptual/Xcode4UserGuide>

Apple Developer Forums

<http://devforums.apple.com>

Related Sessions

Using Interface Building in Xcode 4	Pacific Heights Tuesday 2:00PM
Introducing Interface Builder Storyboarding	Russian Hill Thursday 10:15AM
Full Screen and Aqua Changes	Russian Hill Wednesday 10:15AM
Mastering Source Control in Xcode 4	Nob Hill Wednesday 3:15PM
Mastering Schemes in Xcode 4	Presidio Thursday 9:00AM
Device Management and App Submission with Xcode 4	Presidio Thursday 3:15PM
Effective Debugging with Xcode 4	Pacific Heights Friday 9:00AM

Labs

Xcode 4 Lab

Developer Tools Lab A
Wednesday 11:30AM

Xcode for iOS Development Lab

Developer Tools Lab A
Wednesday 4:30PM

Xcode 4 Lab

Developer Tools Lab A
Thursday 11:30AM





