

iPad

What's New for Software Developers

June 1, 2010

Norman McEntire, Founder
Servin Corporation - <http://servin.com>
Technology Training for Technology Professionals™
norman.mcentire@servin.com

Legal Info

- iPhone and iPad is a trademarks of Apple Inc.
- Servin is a trademark of Servin Corporation.

About Servin Corporation

- Servin **On-Site** Training
 - “Get Your Development Team Up To Speed Fast!”
- Servin **Web-Based** Training
 - “Watch Code Written Before Your Eyes!”
- Servin **Books**
 - Books available on Amazon Kindle and Amazon Kindle Reader (iPhone/iPad, PC, Mac, Blackberry)
- Servin **Apps**
 - 25+ iPhone Apps in various categories

Servin On-Site Training

- “Get Your Development Team Up To Speed Fast!”
- **iPhone** Programming Courses
 - Intro and Advanced
- **Android** Programming Courses
 - Intro and Advanced
- **Linux** Programming Courses
 - Embedded Linux
 - Device Drivers
 - Services/Daemons

Servin Web-Based Training

- “Watch Code Written Before Your Eyes!”
 - Uses Cisco WebEx Player
 - Cisco WebEx Player supported on
 - Linux
 - Mac
 - Windows
- This demo/presentation is also available as Web-Based Training
 - <http://servin.com>

Servin Books

- Books available on [Amazon Kindle](#) and [Amazon Kindle Reader](#) (iPhone, PC, Mac, Blackberry)
 - iPhone [Objective-C 2.0](#) Programming Exercises
 - iPhone [Database](#) Programming Exercises: SQLite
 - iPhone [Camera](#) Programming Exercises
 - iPhone [External Accessory](#) Programming Exercises
 - iPhone Flashlight Programming Tutorial
 - iPhone 3 Programming Templates Explained
 - Red Hat Enterprise Linux 5 Admin Skills
 - SUSE Linux Enterprise Server 11 Admin Skills

Servin Apps

25+ iPhone Apps

- Technology Professionals
 - **Device Info Plus**
 - Device ID Info, Font Info, CPU Info, Network Info, and more!
 - **Accelerometer Fun**
 - Display, record, and export accelerometer info
 - **Process Log**
 - Display list of processes
- Health/Fitness
 - **Weight Log Simple** – keep log of your weight
 - **Sleep Log Simple** – keep log of your sleep
- Games and Entertainment
 - **Touch Fast** – how fast can you touch the screen?
 - **Simon Says Face Up** – can you do as Simon says?

UCSD Extension Courses

- These courses are taught by Norman McEntire and fill up quickly so register early
 - iPhone Programming: Touch, Sound, and More!
 - Summer 2010: July 6 – Aug 31, Tuesdays 6pm - 9pm
 - iPhone Programming: Advanced
 - Summer 2010: July 8 – Sep 2, Thursdays 6pm – 9pm
 - Google's Android
 - Summer 2010: July 10 – Sep 4 , Saturdays 9am - 12pm

Previous SDSIC Presentations

- These SDSIC presentations are available on <http://servin.com>
 - 2010-03-09
 - Mobile Apps 2010: iPhone and Android
 - 2009-07-13
 - iPhone OS 3.0: What's New for Software Developers
 - 2009-04-02
 - iPhone Programming: Touch, Sound, and More!

Opening Remarks

- Welcome!
- Thank you!
- My Promise To You
 - Show you what's new for software developers in iPad
 - iPad => iPhone OS SDK 3.2
- My teaching style
 - Show a few slides
 - Do lots of live demos
 - Questions at anytime are great!

PDF Slides and WebEx Recording Will Be Posted on <http://servin.com>

- My Goal: Show you, ***faster than any method on planet earth***, how to use the new software features for iPad
 - Frameworks. Classes. Methods.
- This demo is fast paced but PDF slides and WebEx recording will be posted on <http://servin.com>
 - The PDF Slides include the source code
 - The WebEx Recording can be paused/replayed

Time for the Demos!

Demo 1

Explore **iPad SDK**

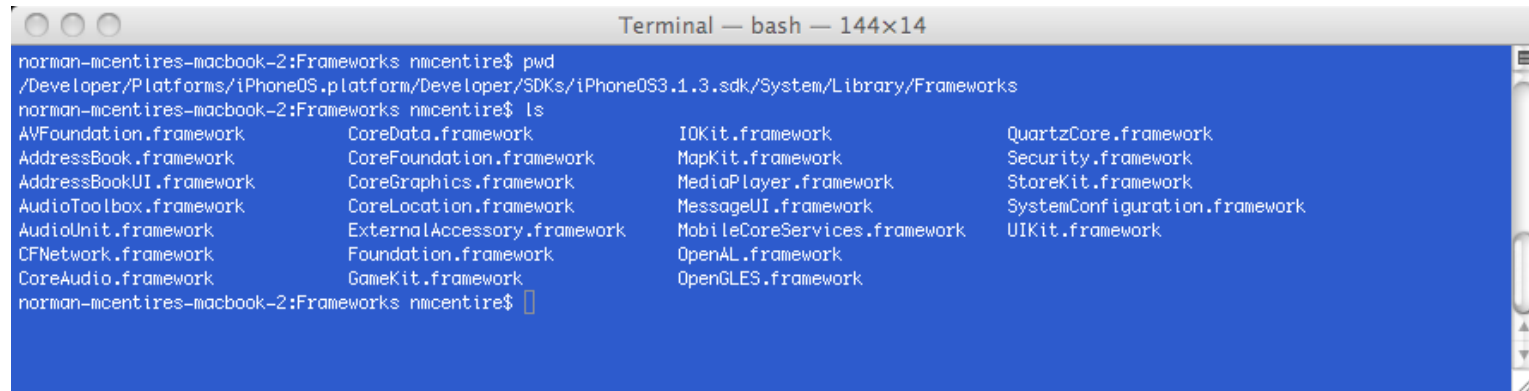
Demo 1 – Key Concepts

- Every iPhone OS SDK provides a specific set of interfaces for use by software developers
- SDK – Software Development Kit
 - iPhone/iPod Touch => iPhone OS 3.1 SDK
 - iPad => iPhone OS 3.2 SDK
- Look for New Frameworks, New Header File Additions, and New Project Templates

Demo 1/Part 1 – New Frameworks

iPhone/iPod Touch Frameworks:

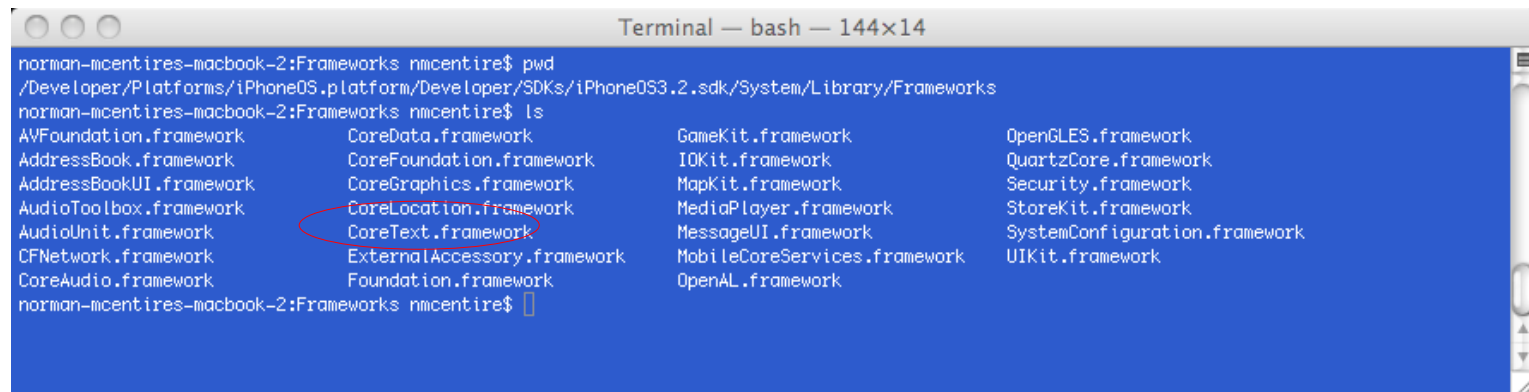
/Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS3.1.3.sdk/System/Library/Frameworks



```
Terminal — bash — 144x14
norman-mcentires-macbook-2:Frameworks nmcentire$ pwd
/Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS3.1.3.sdk/System/Library/Frameworks
norman-mcentires-macbook-2:Frameworks nmcentire$ ls
AVFoundation.framework      CoreData.framework          IOKit.framework            QuartzCore.framework
AddressBook.framework       CoreFoundation.framework   MapKit.framework           Security.framework
AddressBookUI.framework     CoreGraphics.framework     MediaPlayer.framework      StoreKit.framework
AudioToolbox.framework      CoreLocation.framework     MessageUI.framework        SystemConfiguration.framework
AudioUnit.framework         ExternalAccessory.framework MobileCoreServices.framework UIKit.framework
CFNetwork.framework         Foundation.framework        OpenAL.framework           OpenGLES.framework
CoreAudio.framework         GameKit.framework
norman-mcentires-macbook-2:Frameworks nmcentire$
```

iPad Frameworks (New **CoreText.framework**)

/Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS3.2.sdk/System/Library/Frameworks



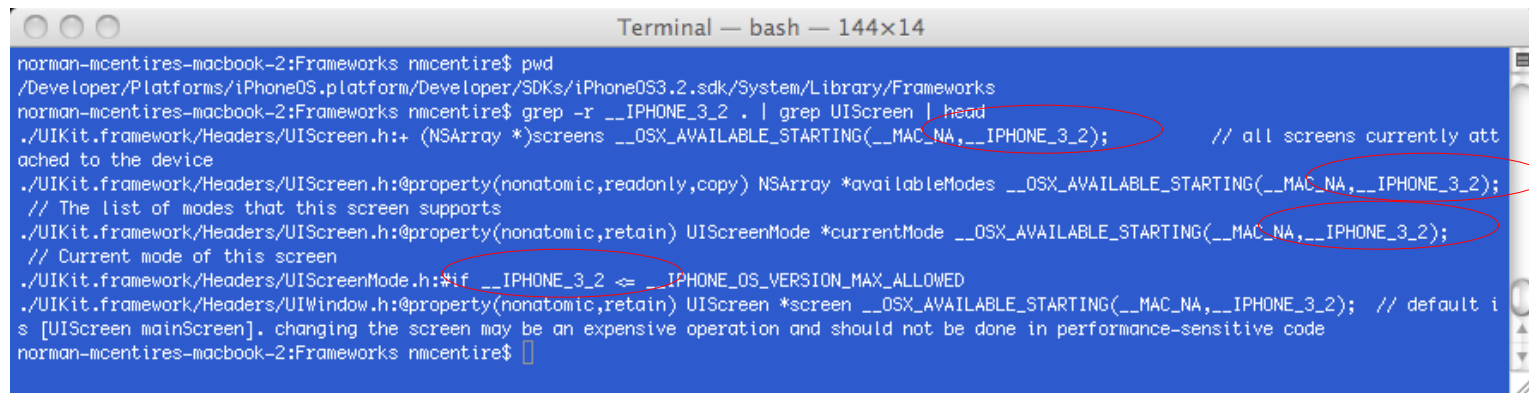
```
Terminal — bash — 144x14
norman-mcentires-macbook-2:Frameworks nmcentire$ pwd
/Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS3.2.sdk/System/Library/Frameworks
norman-mcentires-macbook-2:Frameworks nmcentire$ ls
AVFoundation.framework      CoreData.framework          GameKit.framework          OpenGLES.framework
AddressBook.framework       CoreFoundation.framework   IOKit.framework            QuartzCore.framework
AddressBookUI.framework     CoreGraphics.framework     MapKit.framework           Security.framework
AudioToolbox.framework      CoreLocation.framework     MediaPlayer.framework      StoreKit.framework
AudioUnit.framework         CoreText.framework         MessageUI.framework        SystemConfiguration.framework
CFNetwork.framework         ExternalAccessory.framework MobileCoreServices.framework UIKit.framework
CoreAudio.framework         Foundation.framework        OpenAL.framework
norman-mcentires-macbook-2:Frameworks nmcentire$
```

Demo 1/Part 2 – New Header File Info

iPad New Header File Info (New `__IPHONE_3_2`)

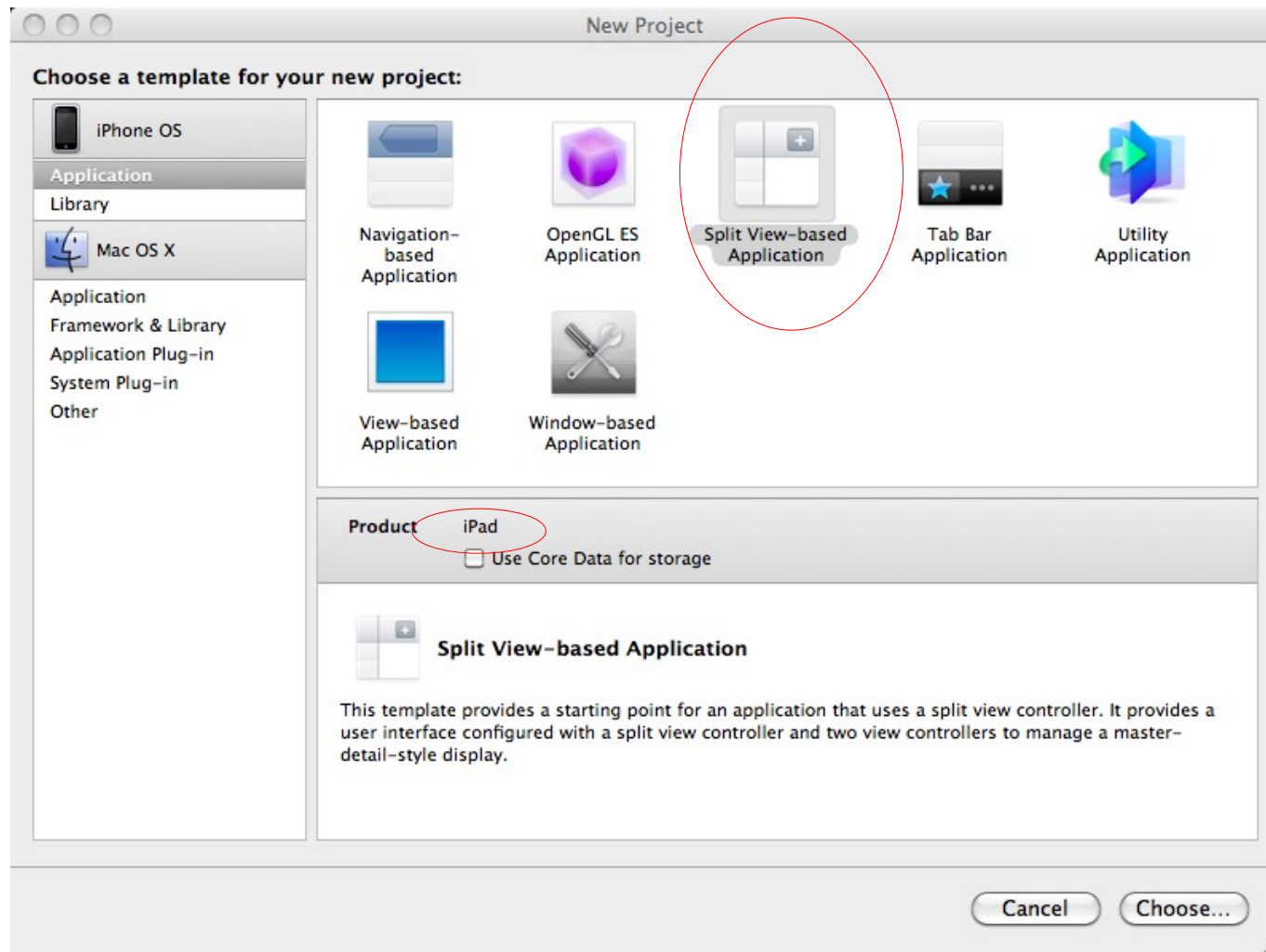
From top-level of Framework Header files, do a **recursive grep** on the pattern `__IPHONE_3_2`

```
$ cd /Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS3.2.sdk/System/Library/Frameworks
$ grep -r __IPHONE_3_2 . | less
```



```
Terminal — bash — 144x14
norman-mcentires-macbook-2:Frameworks nmcentire$ pwd
/Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS3.2.sdk/System/Library/Frameworks
norman-mcentires-macbook-2:Frameworks nmcentire$ grep -r __IPHONE_3_2 . | grep UIScreen | head
./UIKit.framework/Headers/UIScreen.h:+ (NSArray *)screens __OSX_AVAILABLE_STARTING(__MAC_NA,__IPHONE_3_2); // all screens currently attached to the device
./UIKit.framework/Headers/UIScreen.h:@property(nonatomic,readonly,copy) NSArray *availableModes __OSX_AVAILABLE_STARTING(__MAC_NA,__IPHONE_3_2); // The list of modes that this screen supports
./UIKit.framework/Headers/UIScreen.h:@property(nonatomic,retain) UIScreenMode *currentMode __OSX_AVAILABLE_STARTING(__MAC_NA,__IPHONE_3_2); // Current mode of this screen
./UIKit.framework/Headers/UIScreenMode.h:#if __IPHONE_3_2 <= __IPHONE_OS_VERSION_MAX_ALLOWED
./UIKit.framework/Headers/UIWindow.h:@property(nonatomic,retain) UIScreen *screen __OSX_AVAILABLE_STARTING(__MAC_NA,__IPHONE_3_2); // default is [UIScreen mainScreen]. changing the screen may be an expensive operation and should not be done in performance-sensitive code
norman-mcentires-macbook-2:Frameworks nmcentire$
```

Demo 1/Part 3 – New Project Templates



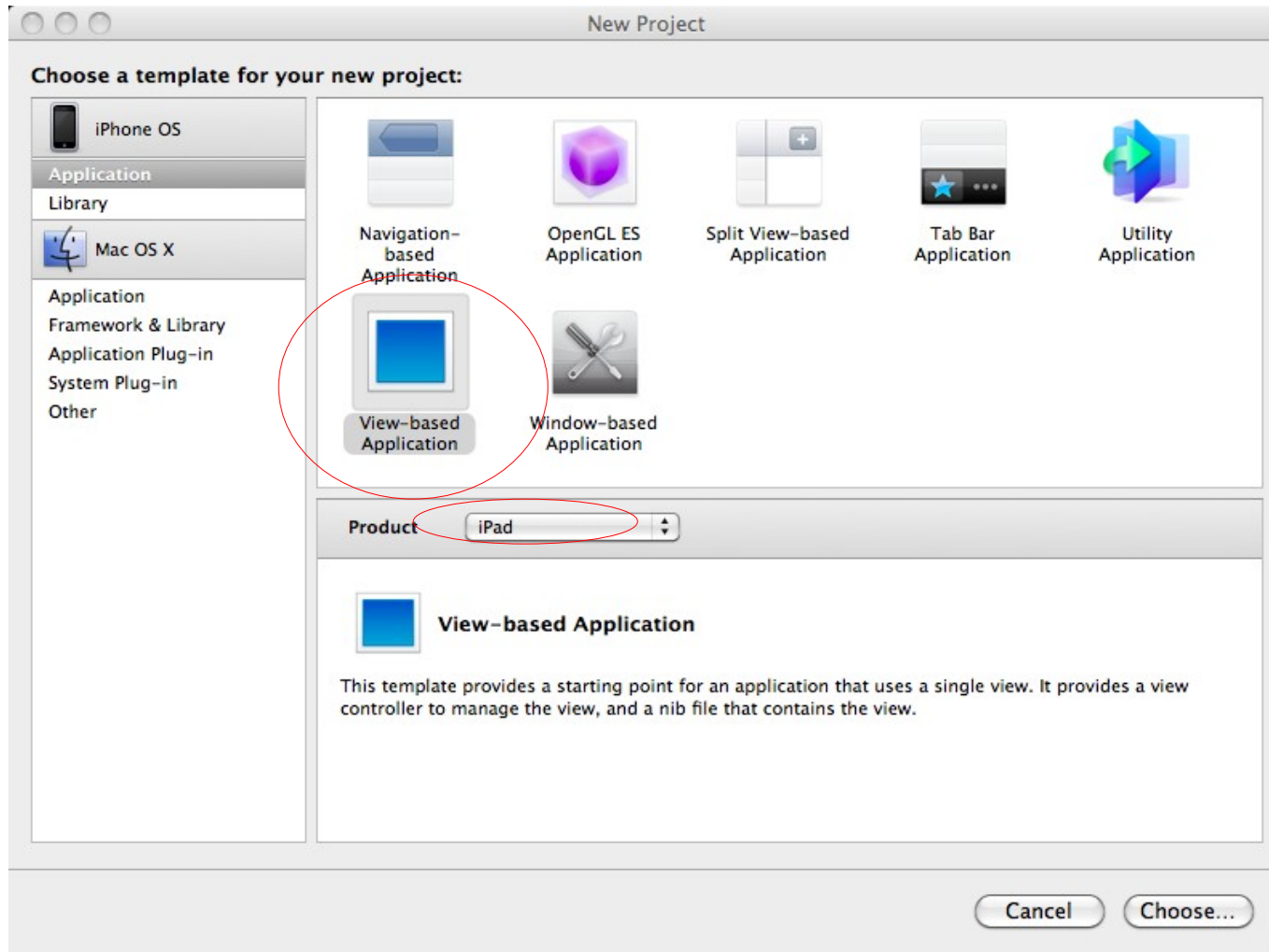
Demo 2

iPad Modal Presentation Styles

Demo 2 – Key Concepts

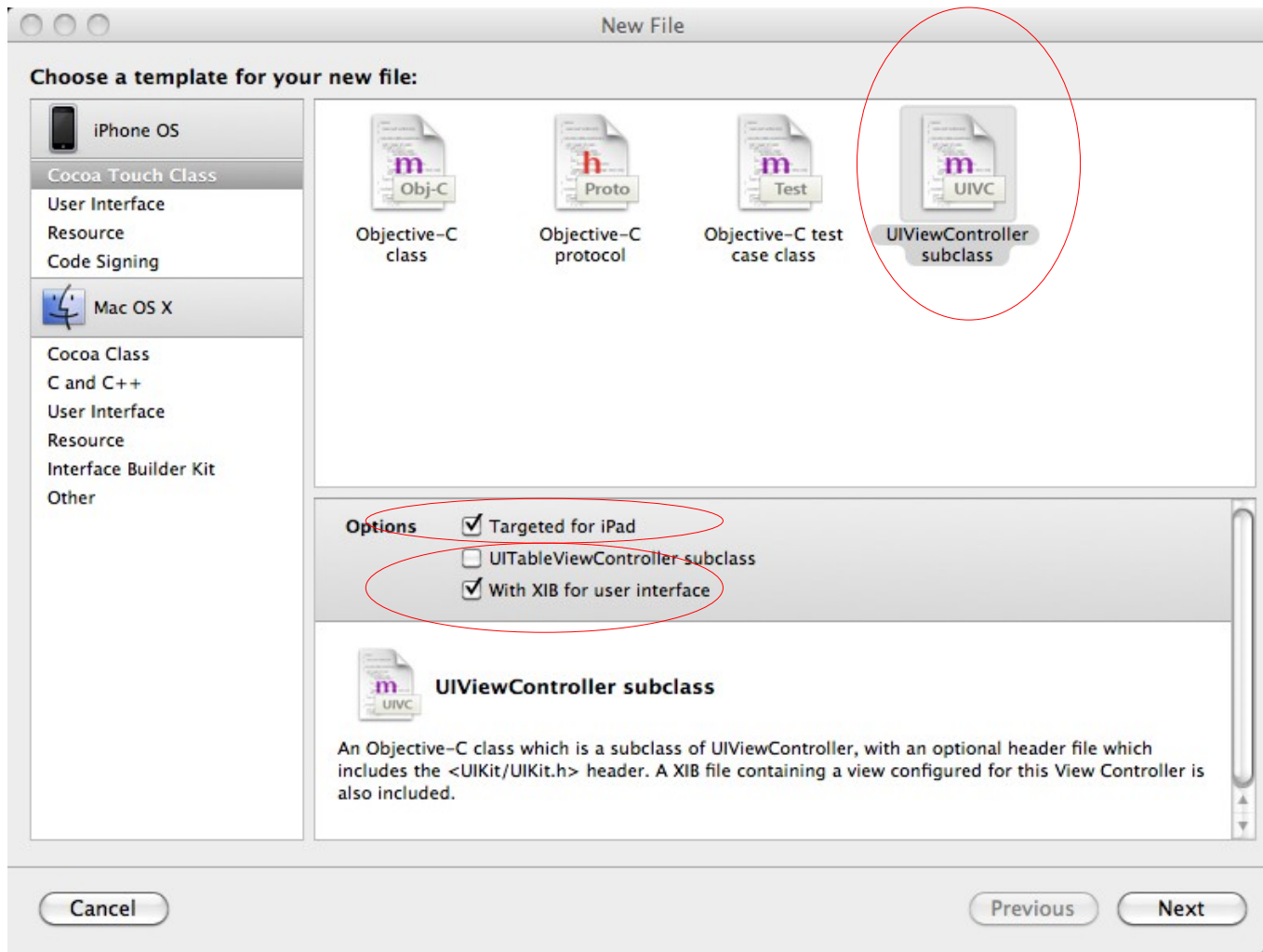
- iPhone/iPod Touch has single fullscreen modal style
- iPad UIViewController provides new **modalPresentationStyle** property
 - Values of modalPresentationStyle Property
 - UIModalPresentationStyleFullScreen
 - UIModalPresentationStylePageSheet
 - UIModalPresentationStyleFormSheet

Demo 2 – Part 1



Demo 2 - Part 2

SecondViewController



Demo 2 - Part 3

FirstViewController.m

```
// FirstViewController.m
```

```
#import "FirstViewController.h"
```

```
#import "SecondViewController.h"
```

```
@implementation FirstViewController
```

```
-(void)touchesBegan:(NSSet *)touches  
                    withEvent:(UIEvent *)event {  
    SecondViewController *vc = [[SecondViewController alloc]  
                                initWithNibName:@"SecondViewController" bundle:nil];  
    vc.delegate = self;  
    //vc.modalPresentationStyle = UIModalPresentationFullScreen;  
    //vc.modalPresentationStyle = UIModalPresentationPageSheet'  
    vc.modalPresentationStyle = UIModalPresentationFormSheet;  
    [self presentModalViewController:vc animated:YES];  
    [vc release];  
}
```

Demo 2 - Part 4

FirstViewController.m

```
- (void)viewDidLoad {  
    [super viewDidLoad];
```

```
    [self.view setBackgroundColor:[UIColor redColor]];
```

```
}
```

Demo 2 - Part 5

SecondViewController.h

```
// SecondViewController.h
#import <UIKit/UIKit.h>
@interface SecondViewController : UIViewController {
    id delegate;
}
@property (nonatomic, assign) id delegate;
@end
```

Demo 2 - Part 6

SecondViewController.m

```
-(void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event {  
    [delegate dismissModalViewControllerAnimated:YES];  
}
```

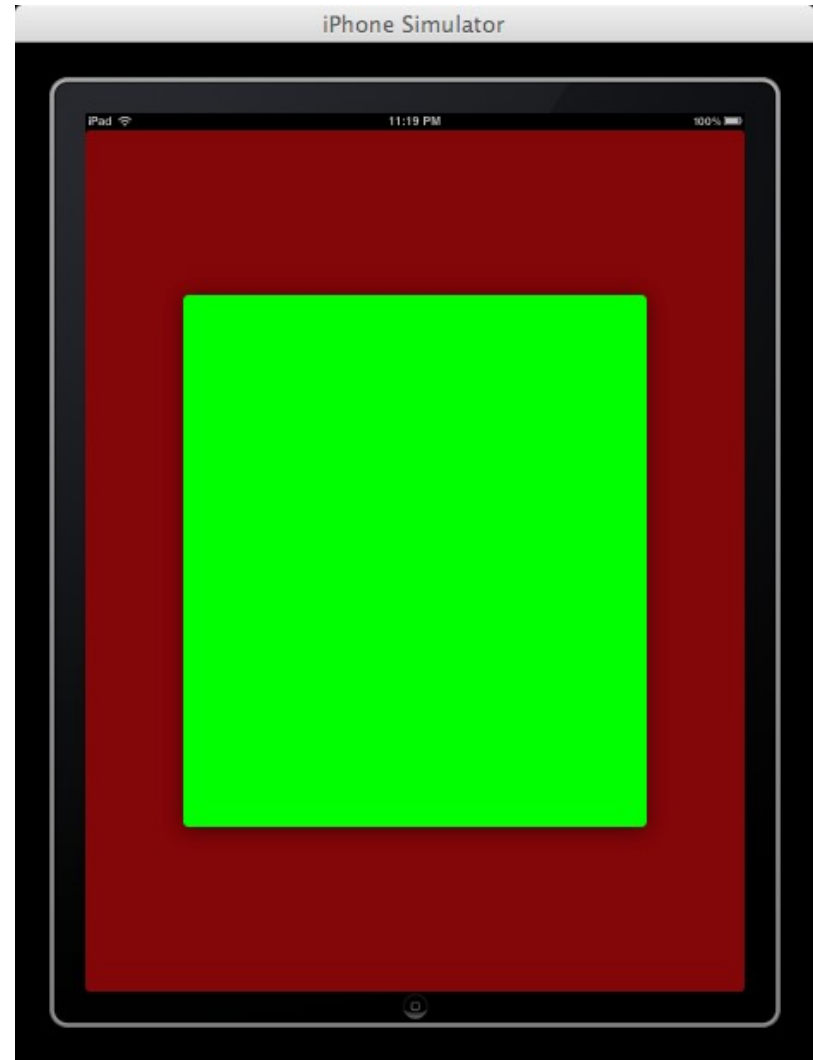
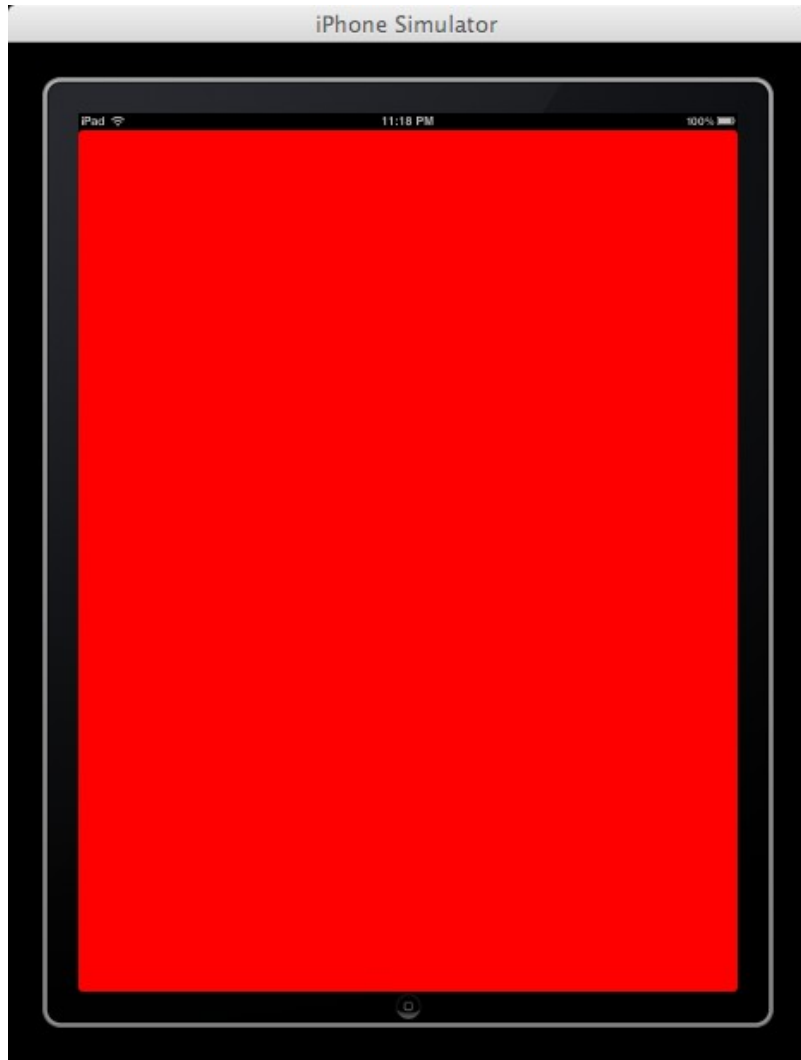
```
- (void)viewDidLoad {  
    [super viewDidLoad];
```

```
    [self.view setBackgroundColor:[UIColor greenColor]];
```

```
}
```

Demo 2 – Results

UIModalPresentationFormSheet



Demo 3

iPad Gesture Recognizers

Demo 3 – Key Concepts

- iPhone OS 3.1 and below had no gesture recognition
- iPad (iPhone OS 3.2) provides gesture recognizers
 - Taps, Long Press (touch and hold), Swipe Left, Swipe Right, Pinching, Rotating
- Gesture Recognizers attach to a view

Demo 3 – New Classes

- UIGestureRecognizer – base class
- UITapGestureRecognizer
- UILongPressGestureRecognizer
- UISwipeGestureRecognizer
- UIPinchGestureRecognizer
- UIPanGestureRecognizer
- UIRotationGestureRecognizer

Demo 3 - Part 1

FirstViewController.m

```
// FirstViewController.m

#import "FirstViewController.h"
#import "SecondViewController.h"

@implementation FirstViewController
//-(void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event {
-(void) doSwipe:(UISwipeGestureRecognizer *)swipe {
    SecondViewController *vc = [[SecondViewController alloc]
        initWithNibName:@"SecondViewController" bundle:nil];
    vc.delegate = self;
    //vc.modalPresentationStyle = UIModalPresentationFullScreen;
    //vc.modalPresentationStyle = UIModalPresentationPageSheet'
    vc.modalPresentationStyle = UIModalPresentationFormSheet;
    [self presentModalViewController:vc animated:YES];
    [vc release];
}
}
```

Demo 3 - Part 2

FirstViewController.m

```
- (void)viewDidLoad {  
    [super viewDidLoad];  
  
    [self.view setBackgroundColor:[UIColor redColor]];
```

```
UISwipeGestureRecognizer *gesture =  
    [[UISwipeGestureRecognizer alloc]  
     initWithTarget:self action:@selector(doSwipe:)];  
gesture.direction = UISwipeGestureRecognizerDirectionRight;  
[self.view addGestureRecognizer:gesture];  
[gesture release];
```

```
}
```

Demo 3 - Part 3

SecondViewController.m

```
//-(void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event {  
-(void) doSwipe:(UISwipeGestureRecognizer *)swipe {  
    [delegate dismissModalViewControllerAnimated:YES];  
}
```

```
- (void)viewDidLoad {  
    [super viewDidLoad];  
  
    [self.view setBackgroundColor:[UIColor greenColor]];
```

```
UISwipeGestureRecognizer *gesture =  
    [[UISwipeGestureRecognizer alloc]  
     initWithTarget:self action:@selector(doSwipe:)];  
gesture.direction = UISwipeGestureRecognizerDirectionLeft;  
[self.view addGestureRecognizer:gesture];  
[gesture release];  
}
```

Demo 4

UIPopoverController

Demo 4 - Concepts

- UIPopoverController displays view controller on top of another
- Similar to UIAlertView, but not modal
 - User does not have to respond to UIPopoverController – touch outside to dismiss
- UIPopoverController displayed where it makes sense
 - Size and “arrow” are configurable
- Example use: Split View-based App

Demo 4 - Part 1

FirstViewController.m

```
// FirstViewController.m

-(void) doSwipe:(UISwipeGestureRecognizer *)swipe {

    SecondViewController *vc =
        [[SecondViewController alloc]
         initWithNibName:@"SecondViewController" bundle:nil];

    UIPopoverController *popover = [[UIPopoverController alloc]
                                     initWithContentViewController:vc];
    vc.delegate = popover;
    CGRect rect = CGRectMake(40.0, 40.0, 100.0, 100.0);
    [popover presentPopoverFromRect:rect inView:self.view
        permittedArrowDirections:UIPopoverArrowDirectionUp
        animated:YES];
    [vc release];
}
```

Demo 4 - Part 2

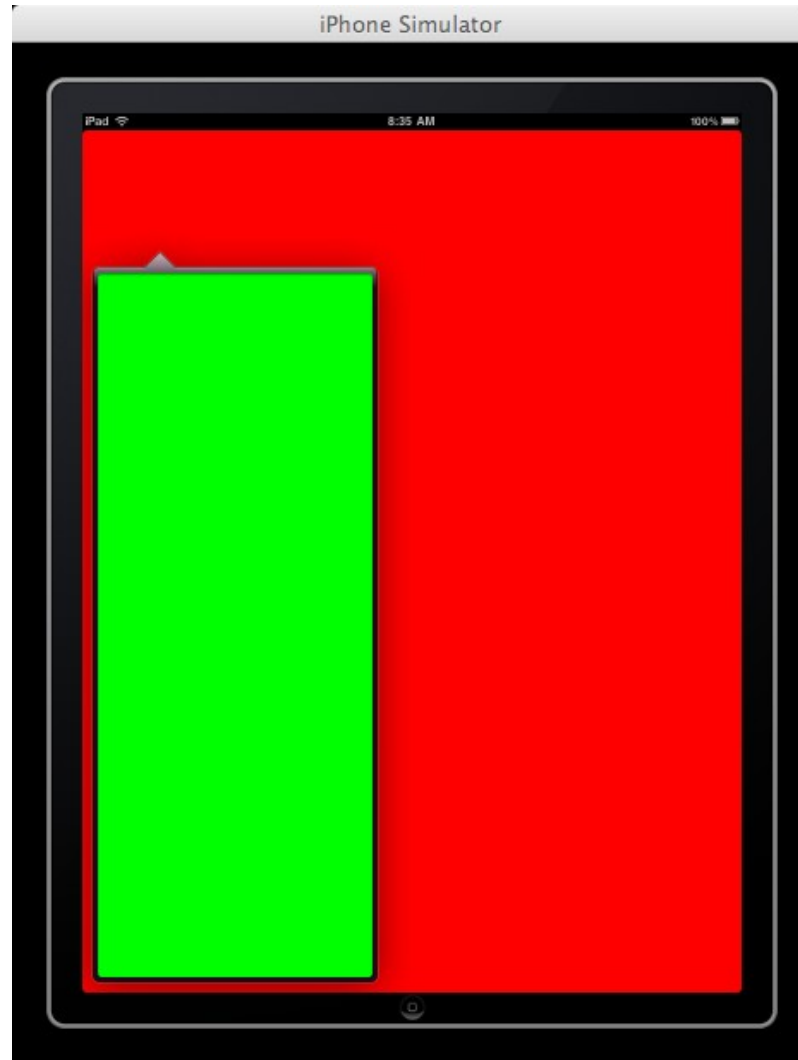
SecondViewController.m

```
-(void) doSwipe:(UISwipeGestureRecognizer *)swipe {
```

```
[delegate dismissPopoverAnimated:YES];
```

```
}
```

Demo 4 - Results



Demo 5

Keyboard Accessory

Demo 5 – Key Concepts

- iPad provides for custom input views and custom accessory views
 - Applies to UITextField and UITextView
- New UIResponder Properties
 - inputView
 - inputAccessoryView

Demo 5 - Part 1

FirstViewController.h

```
// FirstViewController.h

#import <UIKit/UIKit.h>

@interface FirstViewController : UIViewController {

    IBOutlet UITextField *myTextField;

}

@end
```

Demo 5 - Part 2

FirstViewController.m

```
// FirstViewController.m
```

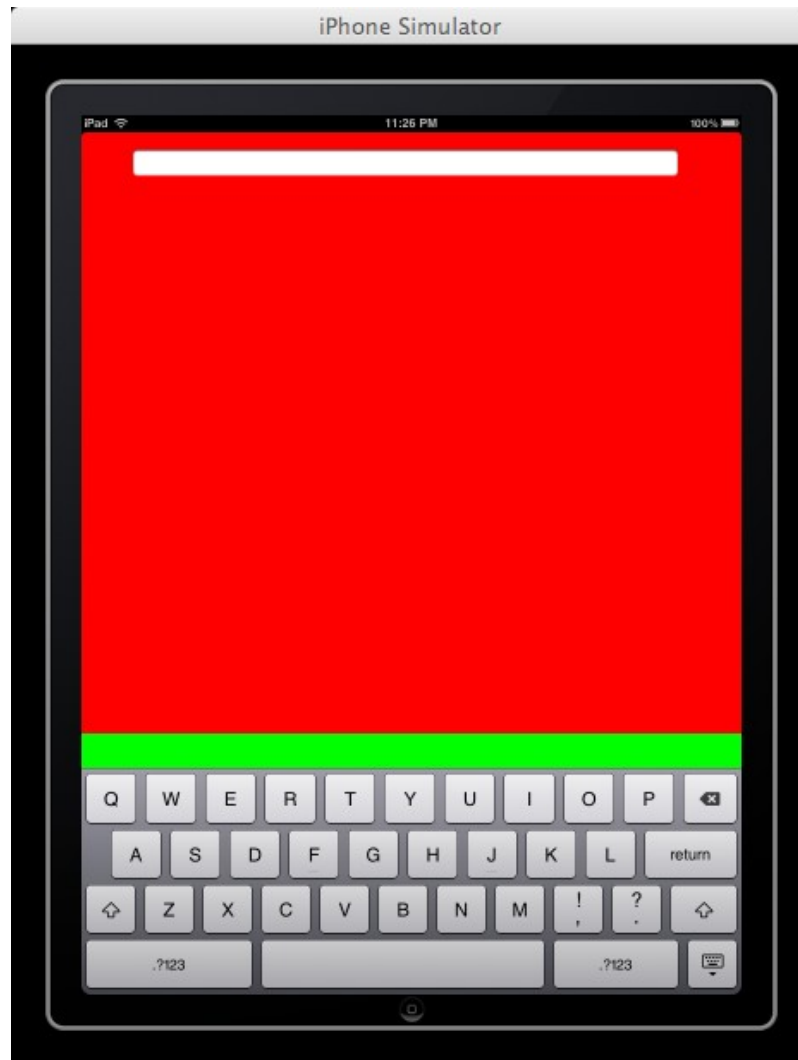
```
...
```

```
- (void)viewDidLoad {  
    [super viewDidLoad];
```

```
    CGRect rect = CGRectMake(0.0, 0.0, 40.0, 40.0);  
    UIView *myView = [[UIView alloc] initWithFrame:rect];  
    [myView setBackgroundColor:[UIColor greenColor]];  
    myTextField.inputAccessoryView = myView;
```

```
...
```

Demo 5 – Results



Demo 6

iPad Screens

Demo 6 – Concepts

- iPad has larger screen size
 - iPhone/iPod Touch: 320 x 480
 - iPad: 768 x 1024
- iPad has **two screens** (internal and external)
- Use notification center to detect when external screen connected/disconnected
 - **UIScreenDidConnectNotification**
 - **UIScreenDidDisconnectNotification**

Demo 6 – FirstViewController.m

```
-(void)doScreenConnect:(NSNotification *)notification {
    NSLog(@"%@", [notification description]);
}
- (void)viewDidLoad {
    [super viewDidLoad];

    NSArray *screens = [UIScreen screens];
    for (UIScreen *screen in screens) {
        NSLog(@"screen.bounds: width: %1.0f, height: %1.0f",
            screen.bounds.size.width,
            screen.bounds.size.height);
    }

    [[NSNotificationCenter defaultCenter]
        addObserver:self
        selector:@selector(doScreenConnect:)
        name:UIScreenDidConnectNotification
        object:nil];
}
```

...

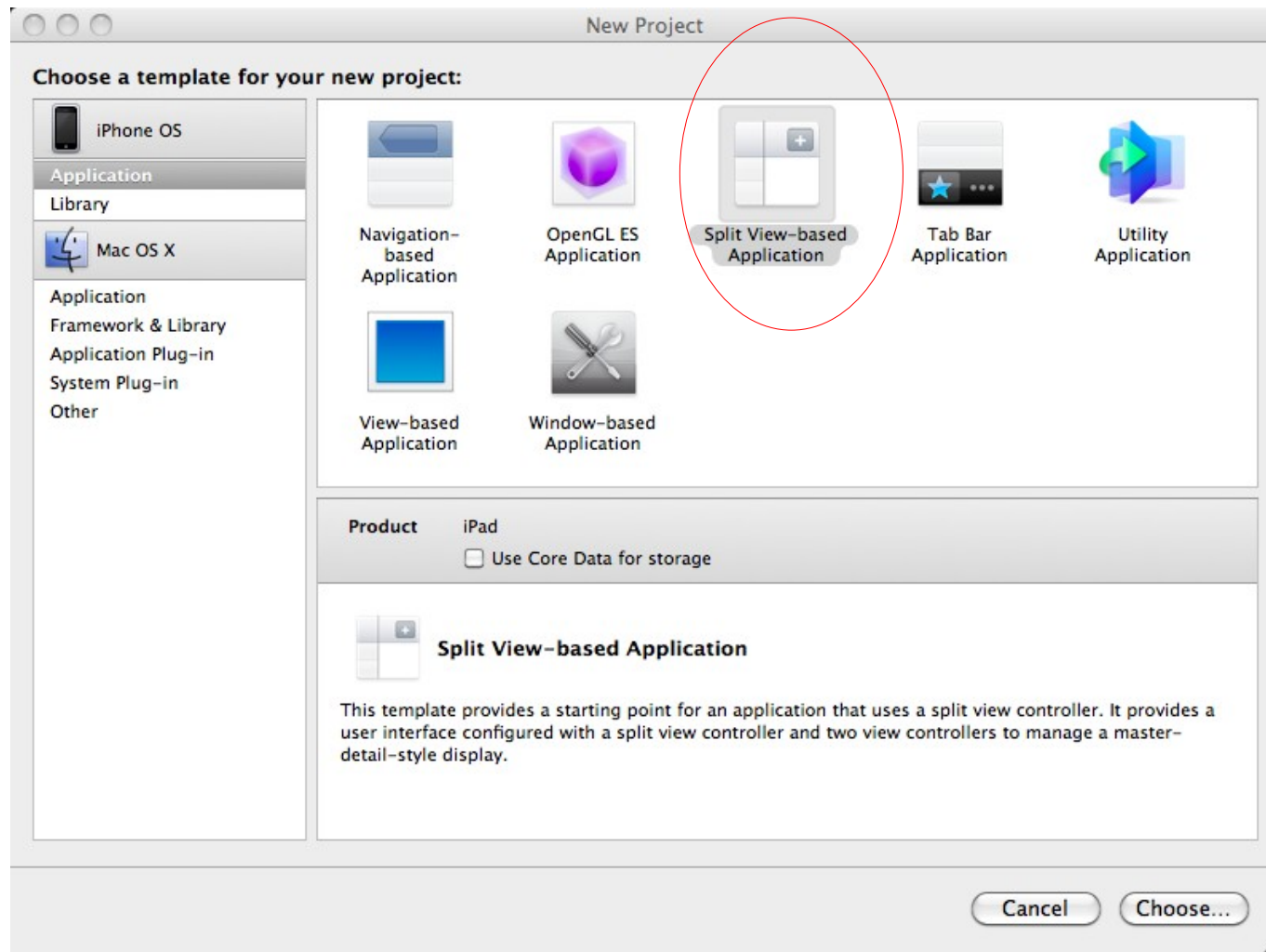
Demo 7

Split View-based Application

Demo 7 – Concepts

- iPad introduces **UISplitViewController** to manage two view controllers
 - RootViewController, DetailViewController
- **Auto hides RootViewController** on Portrait; Auto displays RootViewController on Landscape
- iPad introduces **UIPopoverController** to display “hidden” RootViewController

Demo 7/Part 1 – Split View-based App



Demo 7/Part 2 - HelloAppDelegate.h

```
// HelloAppDelegate.h

#import <UIKit/UIKit.h>

@class RootViewController;
@class DetailViewController;

@interface HelloAppDelegate : NSObject <UIApplicationDelegate> {

    UIWindow *window;

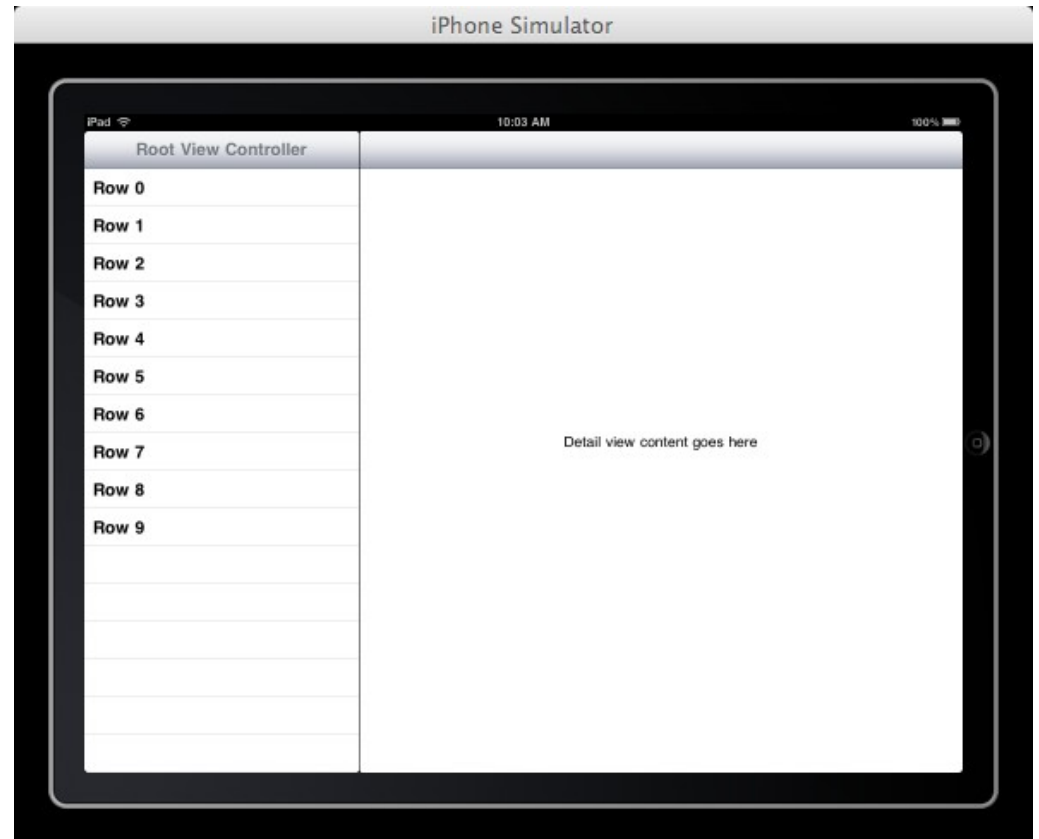
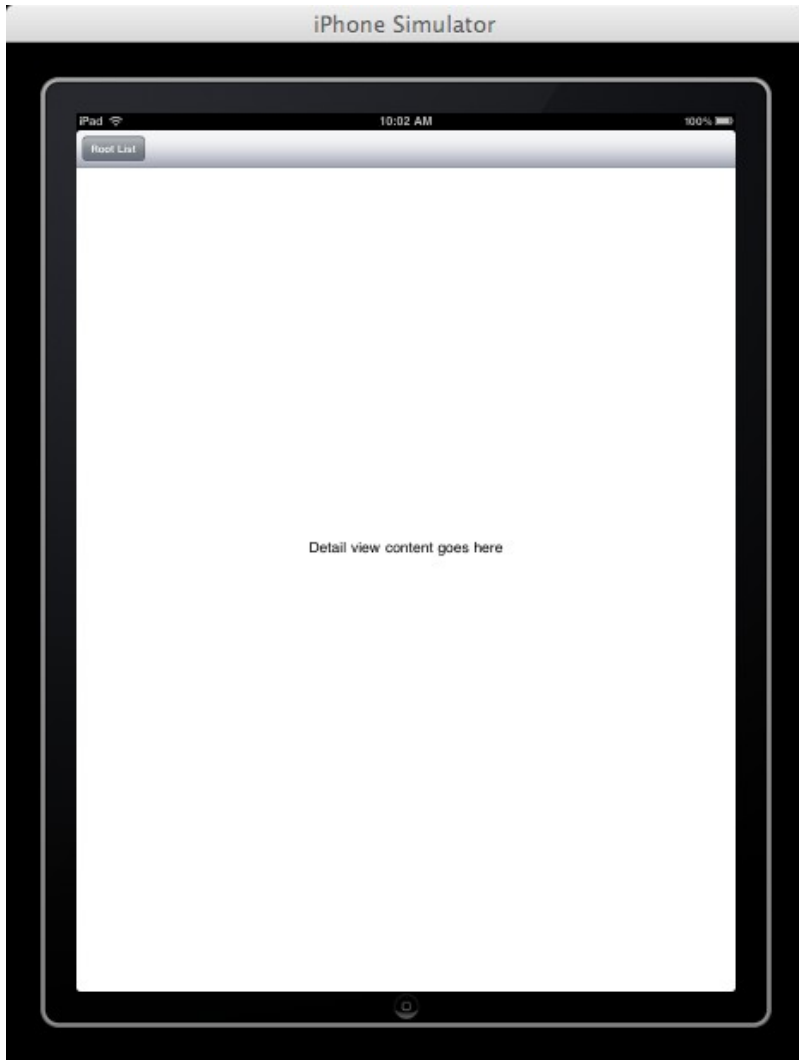
    UISplitViewController *splitViewController;

    RootViewController *rootViewController;
    DetailViewController *detailViewController;
}

...

```

Demo 7/Part 3



Demo 8. PDF Generation

Demo 8 - Concepts

- Ipad provides generation of PDF content
 - Generate as either data or into file
- The overall flow
 - UIGraphicsBeginPDFContextToFile(...)
 - UIGraphicsBeginPDFPage()
 - Draw content using Core Graphics and Core Text
 - UIGraphicsEndPDFContext()

Demo 8 - FirstViewController.h

```
// M1ViewController.h
```

```
#import <UIKit/UIKit.h>
```

```
@interface M1ViewController : UIViewController {  
    IBOutlet UITextField *myTextField;
```

```
    IBOutlet UIWebView *myWebView;
```

```
}
```

```
-(IBAction) generatePDF;  
-(IBAction) displayPDF;
```

```
@end
```

Demo 8 - FirstViewController.m

```
-(IBAction) generatePDF {  
  
    // Pathname to file  
    NSString *path = [NSHomeDirectory()  
        stringByAppendingPathComponent:@"Documents/demo.pdf"];  
  
    // Create PDF context to file  
    CGRect bounds = CGRectZero; //Default 8.5x11 page  
    BOOL success =  
        UIGraphicsBeginPDFContextToFile (path, bounds, nil);  
  
    if (success != YES) {  
        NSLog(@"no success");  
        return;  
    }  
}
```

Demo 8 - FirstViewController.m

```
// Start a new page
UIGraphicsBeginPDFPage();

// Get current date/time
NSString *dateTime = [[NSDate date] description];

// Prepare the text
CFAttributedStringRef dateTimeText =
CFAttributedStringCreate(NULL, (CFStringRef)dateTime, NULL);
CTFramesetterRef framesetter =
CTFramesetterCreateWithAttributedString(dateTimeText);
if (framesetter == NULL) {
    NSLog(@"todo");
    return;
}
```

Demo 8 - FirstViewController.m

```
// Get the graphics context.
CGContextRef context = UIGraphicsGetCurrentContext();

// Put the text matrix into a known state. This ensures
// that no old scaling factors are left in place.
CGContextSetTextMatrix(context, CGAffineTransformIdentity);

// Create a path object to enclose the text. Use 72 point
// margins all around the text.
CGRect frameRect = CGRectMake(72, 72, 468, 648);
CGMutablePathRef framePath = CGPathCreateMutable();
CGPathAddRect(framePath, NULL, frameRect);
```

Demo 8 - FirstViewController.m

```
// Get the frame that will do the rendering.
CFRange currentRange = CFRangeMake(0, 0);
CTFrameRef frameRef =
CTFramesetterCreateFrame(framesetter, currentRange, framePath,
NULL);
CGPathRelease(framePath);

// Core Text draws from the bottom-left corner up, so flip
// the current transform prior to drawing.
CGContextTranslateCTM(context, 0, 792);
CGContextScaleCTM(context, 1.0, -1.0);

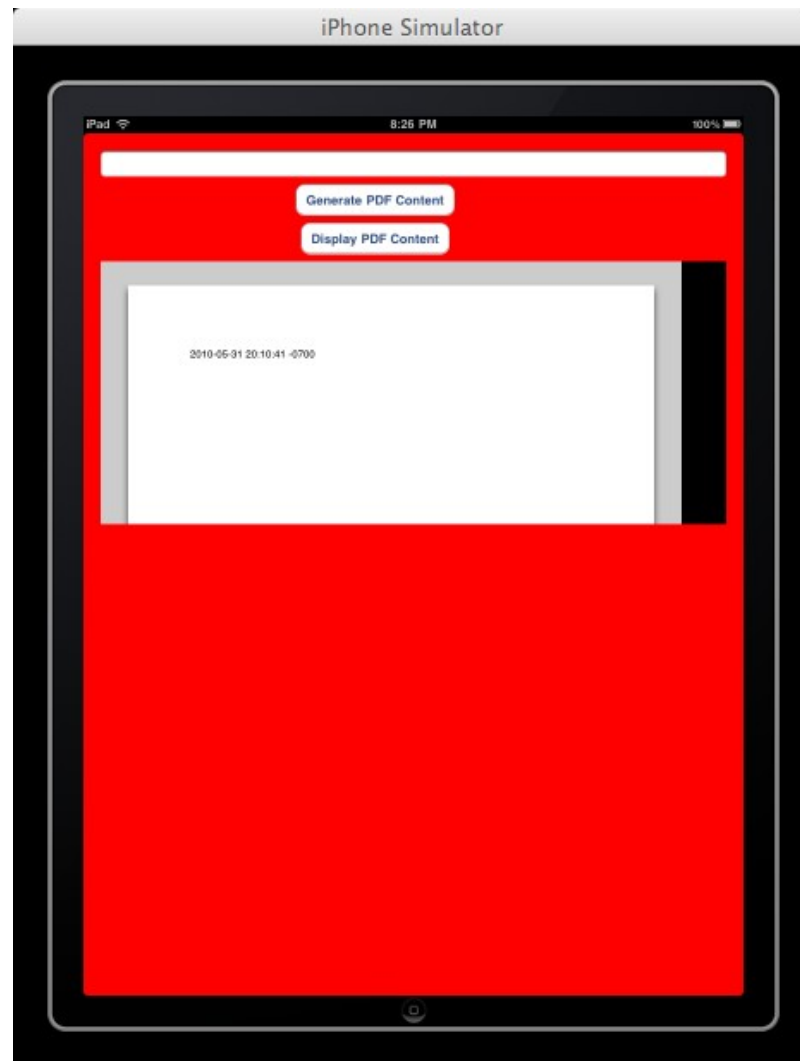
// Draw the frame.
CTFrameDraw(frameRef, context);

// All done with page
UIGraphicsEndPDFContext();
}
```

Demo 8 - FirstViewController.m

```
-(IBAction) displayPDF {  
    // Pathname to file  
    NSString *path =  
        [NSHomeDirectory()  
         stringByAppendingPathComponent:@"Documents/demo.pdf"];  
  
    NSURL *url = [NSURL URLWithString:path];  
    NSURLRequest *request = [NSURLRequest requestWithURL:url];  
    [myWebView loadRequest:request];  
}
```

Demo 8 - Results



Questions/Answers

Thank You!