

# Getting started with Provisioning and Debugging with XCode

Salik Syed

CS193P Spring 2011



# Part 1: Running apps on a device

- Apple has strict security regarding who can distribute apps
- Each app that is compiled to run on a device must be “code-signed”



# Two things are necessary

- Developer Certificate: Verifies your identity
- Provisioning profile: Specifies application and device permissions

- You should have already received an invitation to the Apple Developer program
- You will need to login to the portal to obtain your Developer Certificate
- We will use a shared CS193P provisioning profile which will already have your device UDID in it.

# Step 1: login to the provisioning portal

- <http://developer.apple.com>
- Click “iOS Dev Center”
- Login using the developer account you created
- Click iOS Provisioning profile on right menu
- **Make sure you use Safari and not another browser. The portal glitches in other browsers**

## iOS Provisioning Portal

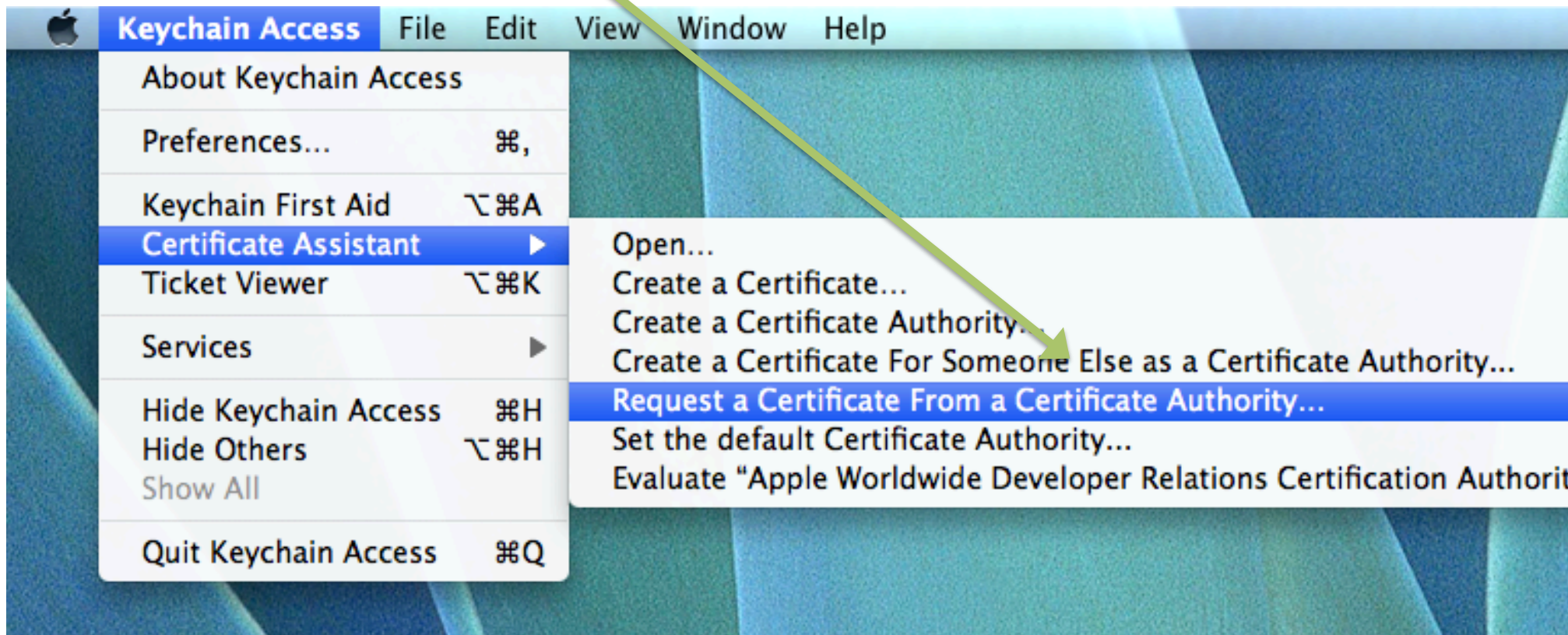
The screenshot shows the Apple Developer website interface. At the top, there is a navigation bar with the Apple logo and the word "Developer". To the right of the logo are links for "Technologies", "Resources", "Programs", "Support", and "Member Center". A search bar labeled "Search Developer" is also present. Below the navigation bar is a dark header with "iOS Dev Center" and sub-sections for "iOS Dev Center", "Mac Dev Center", and "Safari Dev Center". A user greeting "Hi, Salik Syed" is visible, along with links for "My Profile" and "Log out".

The main content area is divided into several sections:

- Developing for iOS 4**: A section header.
- Resources for iOS 4.3**: A section with two items:
  - Downloads**: "Download the latest build of iOS SDK" with a download icon.
  - Getting Started Videos**: "Watch Apple experts discuss a range of" with a video icon.
- Featured Content**: A section with three items:
  - New Subscription Service for iOS Apps**
  - Getting Ready for iOS 4.3**
  - What's New in iOS 4**
- iOS Developer Program**: A menu with three items:
  - iOS Provisioning Portal** (highlighted with a green arrow)
  - Apple Developer Forums**
  - Developer Support Center**

# Step 2: Create a certificate Request

- Open “Keychain Access” Application
- Choose the following from top menu



Certificate Assistant

### Certificate Information

Enter information for the certificate you are requesting.  
Click Continue to request a certificate from the CA.

User Email Address:

Common Name:

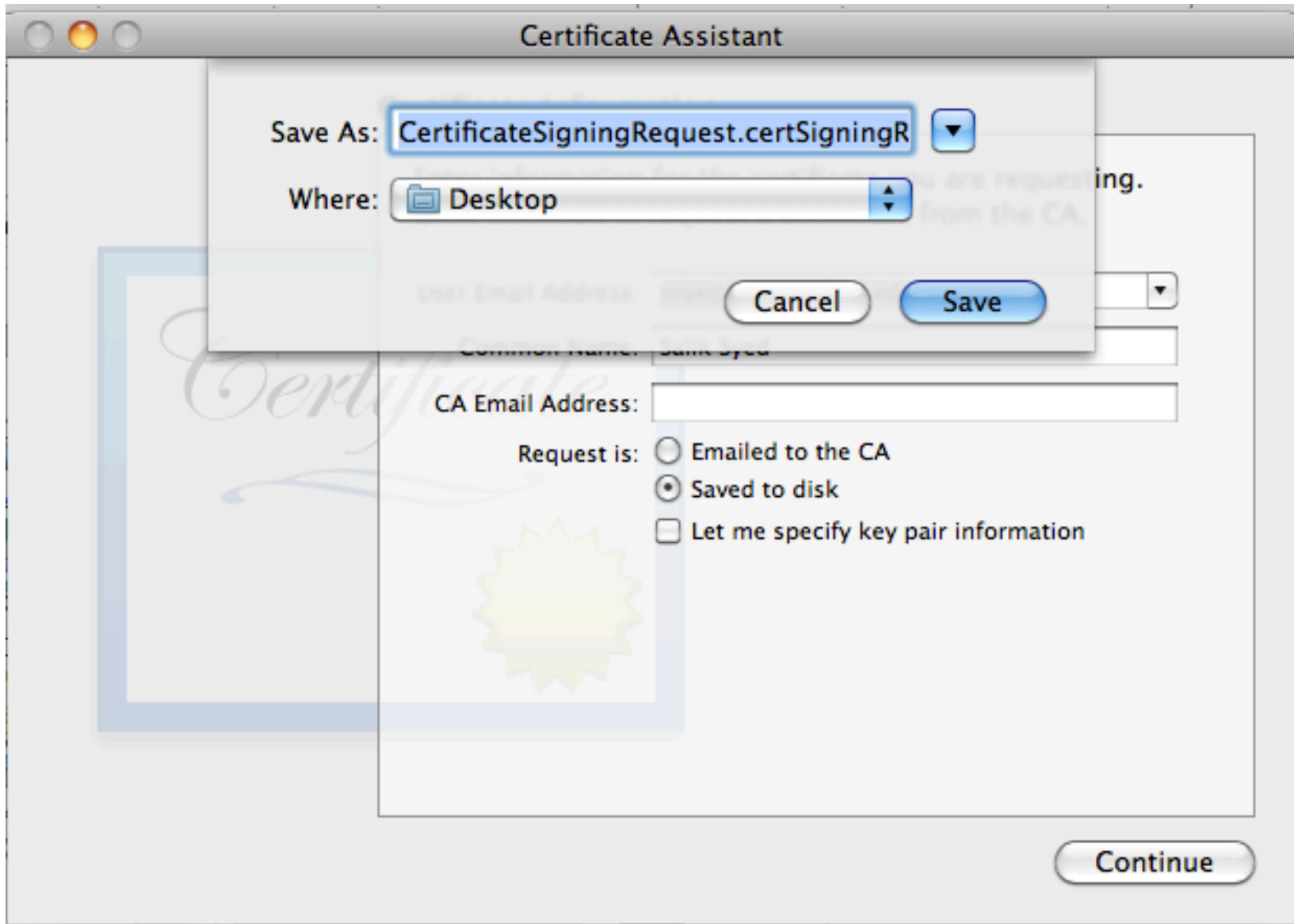
CA Email Address:

Request is:  Emailed to the CA  
 Saved to disk  
 Let me specify key pair information

Continue

Ensure that "Saved to disk" option is chosen.  
Use the same name and email that you used for your Developer account





## Step 3: Submit certificate request

- Go back to the iOS developer portal site
- Click “Certificates” on left menu
- Click “Request Certificate”

Certificates Menu Link

Request Certificate

iOS Provisioning Portal

Welcome, Salik Syed | Edit Profile | Log out

Provisioning Portal : Stanford University (President/Provost) [Go to iOS Dev Center](#)

Home  
**Certificates**  
Devices  
App IDs  
Provisioning

Development History How To

### Current Development Certificates

**Your Certificate**

Name	Provisioning Profiles	Expiration Date	Status	Action
ⓘ You currently do not have a valid certificate				<a href="#">Request Certificate</a>


\*If you do not have the WWDR intermediate certificate installed, [click here to download now](#).


- Choose the file you saved earlier in certificate assistant.
- Press Submit

Yours may say "Pending Issuance" check back in a day and it should change status. (We need to approve all requests)


Development History How To

### Current Development Certificates

 **Your Certificate**

Name	Provisioning Profiles	Expiration Date	Status	Action
 Salik Syed		Apr 06, 2012	Issued	<a href="#">Download</a> <a href="#">Revoke</a>

\*If you do not have the WWDR intermediate certificate installed, [click here to download now](#).

 **Team Signing Requests (1)**

Also download the WWDR intermediate certificate



\*If you do not have the WWDR intermediate certificate installed, [click here to download now.](#)

## Step 4: Install Certificates

- Once your certificate has changed status to “Issued” you can click download
- First double click the WWDR certificate to install that -- it will bring up keychain access
- Once you have saved your certificate double click it and it will bring up Keychain Access

Keychain Access

Click to lock the login keychain.

Keychains

- login
- System
- System Roots

Category

- All Items
- Passwords
- Secure Notes
- My Certificates
- Keys
- Certificates

**iPhone Developer: Salik Syed (S2EU9RC5S7)**  
Issued by: Apple Worldwide Developer Relations Certification Authority  
Expires: Friday, April 6, 2012 2:51:30 PM PT  
✔ This certificate is valid

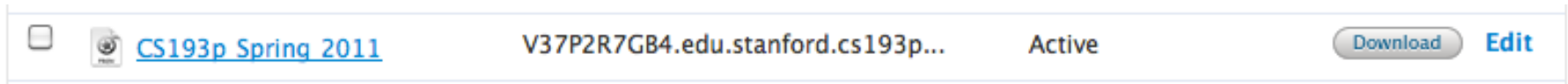
Name	Kind	Expires
Apple Worldwide Developer Relations Certification Authority	certificate	Feb 14, 2016 10:00 AM PT
iPhone Developer: Salik Syed (7BL3KQNY4Y)	certificate	Dec 10, 2011 3:00 PM PT
Salik Syed	private key	--
iPhone Developer: Salik Syed (S2EU9RC5S7)	certificate	Apr 6, 2012 2:51:30 PM PT
Salik Syed	private key	--
iPhone Distribution: Draper Fisher Jurvetson Management, LLC	certificate	Jan 29, 2012 8:20 AM PT
iPhone Distribution: Salik Syed	certificate	Dec 13, 2011 9:00 AM PT

Make sure you have a dropdown arrow near "iPhone Developer". When you click it you should see your name along with a key icon



# Step 5: Download Provisioning Profile

- Click “Provisioning” on left menu
- Download the CS193p Spring 2011 Profile.
- You **MUST** have submitted your UDID to get this to work.




## Step 6: Add provisioning profile to XCode

- Double click the downloaded profile.
- It should pull up the organizer in XCode

# You did something wrong if...

Xcode could not find a valid private-key/certificate pair for this profile in your keychain.



### CS193p Spring 2011

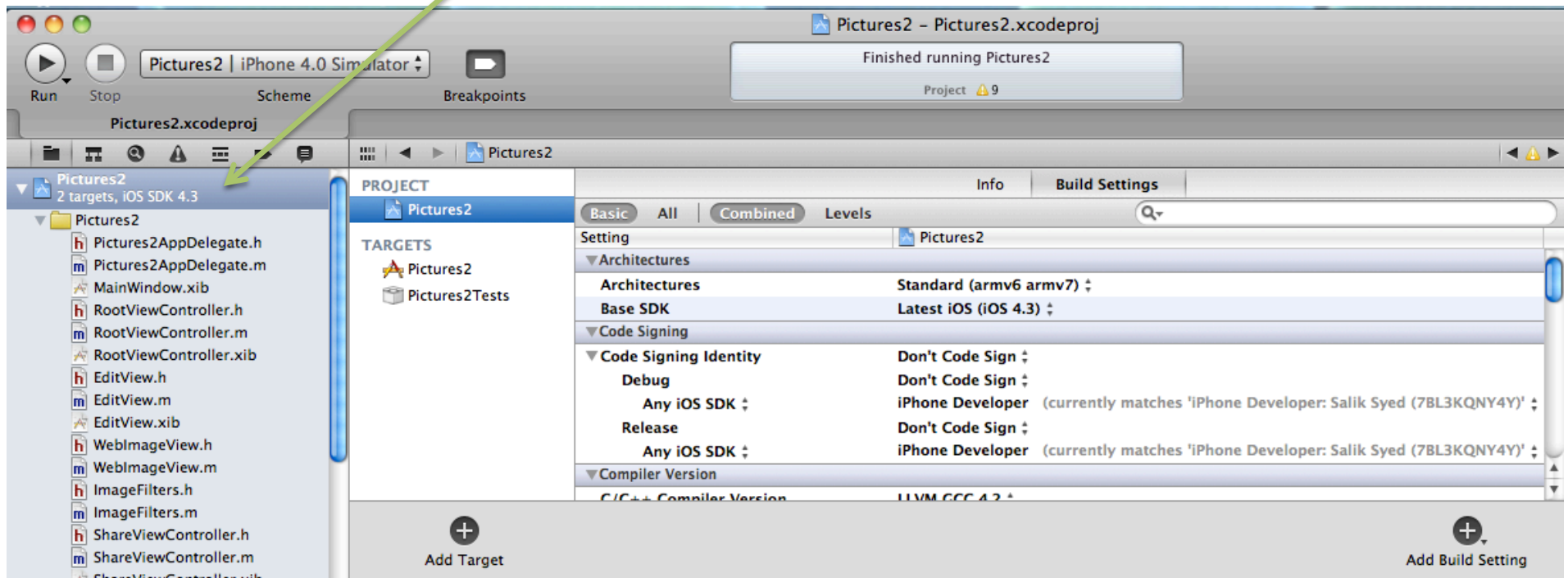
Creation Date Saturday, March 26, 2011 10:08:45 PM PT  
Expiration Date Friday, June 24, 2011 10:08:45 PM PT  
Profile Identifier 865D0949-9177-4294-AE19-493846A48AEF  
App Identifier V37P2R7GB4.edu.stanford.cs193p.spring2011  
Devices 3 devices

Name	Expiration Date	Status
All	May 22, 2011 4:54 PM	
AppStore2	December 12, 2011 9:34 PM	
CS193p Spring 2011	June 24, 2011 10:08 PM	⚠ Valid signing identity not found
DFJ Distribution	January 28, 2012 8:33 AM	
Team Provisioning Profile: *	June 30, 2011 5:39 PM	
iRaveInApp	June 21, 2011 12:47 PM	
iRaveTest	May 24, 2011 11:43 AM	

Import Export  Automatic Device Provisioning Refresh

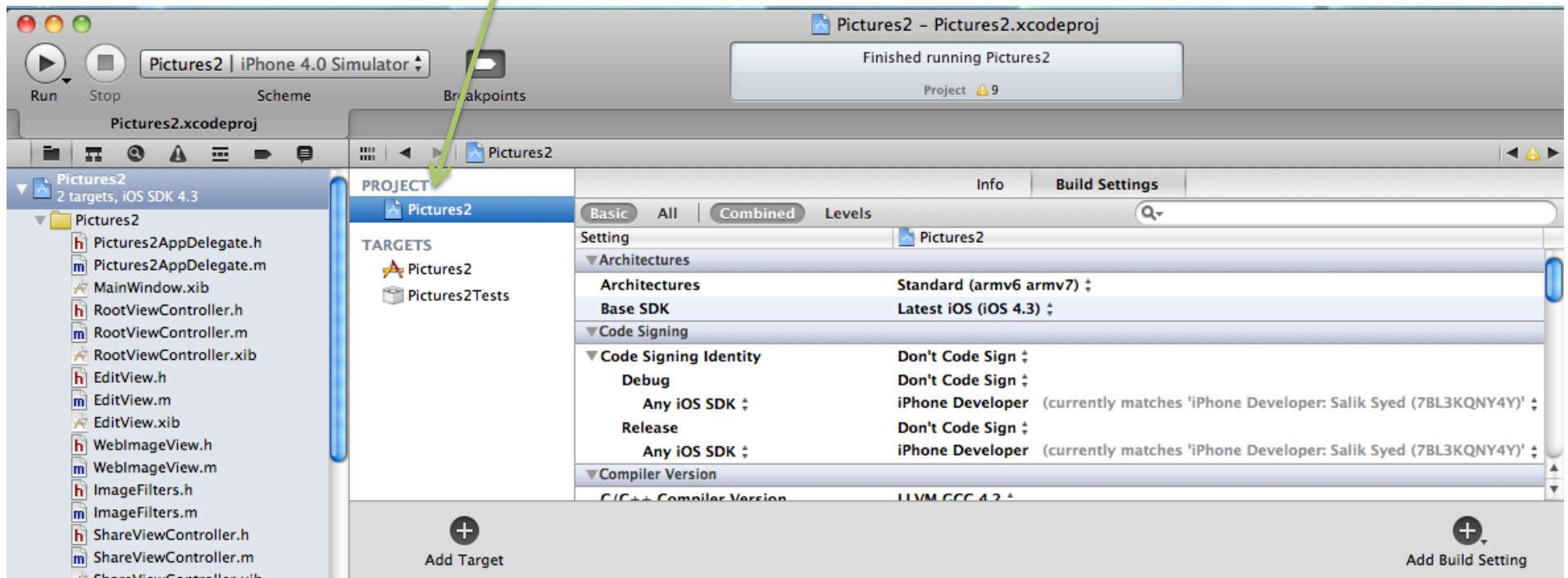
# Step 7 : Setup project to run on iOS Device

Click the root Project on navigator



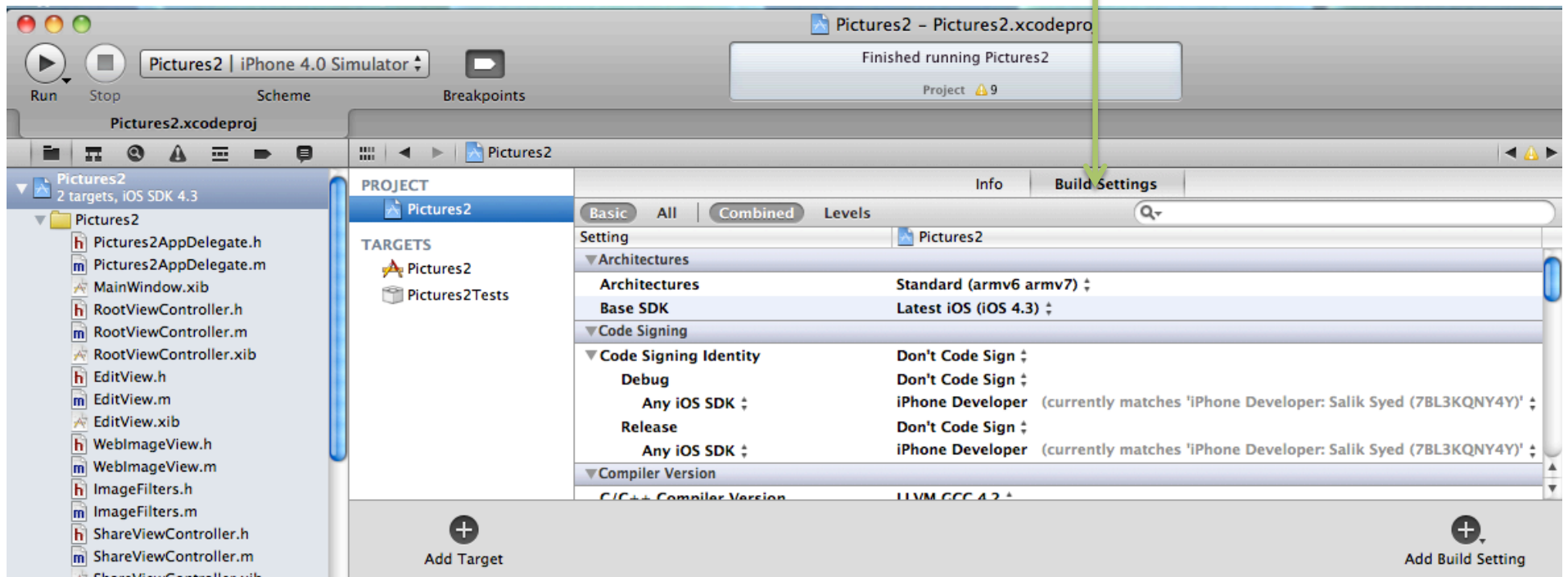
# Step 7 : Setup project to run on iOS Device

Click your project under "Projects Heading"



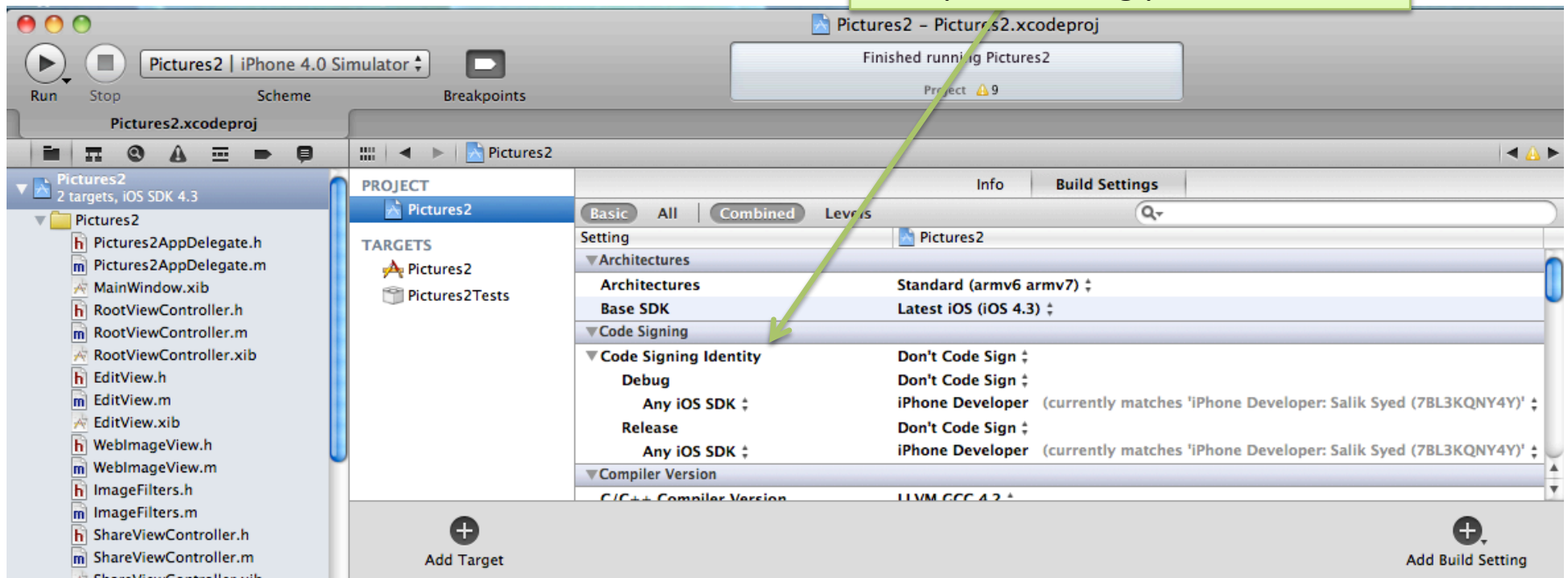
# Step 7 : Setup project to run on iOS Device

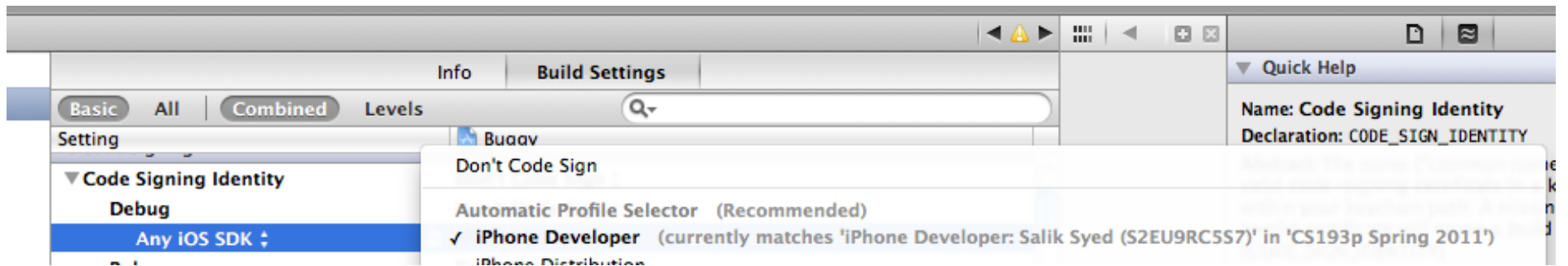
Go to build settings Tab



# Step 7 : Setup project to run on iOS Device

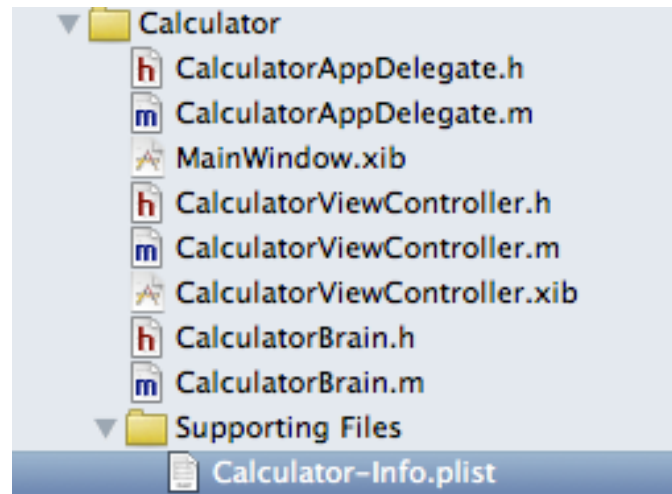
This is where you choose identity and provisioning profile







# Click the .plist file in browser



# Step 8: Edit Target App Id to be the same as your provisioning profile

**edu.stanford.cs193p.spring2011**

Localization native development region	String	en
Bundle display name	String	\${PRODUCT_NAME}
Executable file	String	\${EXECUTABLE_NAME}
Icon file	String	
Bundle identifier	String	edu.stanford.cs193p.spring2011
InfoDictionary version	String	6.0
Bundle name	String	\${PRODUCT_NAME}
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle creator OS Type code	String	????
Bundle version	String	1.0
Application requires iPhone environment	Boolean	YES
Main nib file base name	String	MainWindow
▶ Supported interface orientations	Array	(1 item)

# That's it!

- Make sure you have submitted your UDID to us.
- Plug in your iOS device and select it from the drop down near the “Run” button
- If you are using a device with older than 4.3 iOS then change the deployment target in build settings to the proper device (or else it will not work!)

# Part 2: Debugging with XCode

- We will step through 3 Common bugs



```
//  
  
#import "BuggyViewController.h"  
  
@interface BuggyViewController()  
@property (readonly) NSMutableArray * arrayOfStrings;  
@end  
  
@implementation BuggyViewController  
  
@synthesize arrayOfStrings,arrayContentsLabel;  
  
-(void) updateDisplay  
{  
    arrayContentsLabel.text = [arrayOfStrings objectAtIndex:0];  
}  
  
-(IBAction)addObject:(id)sender  
{  
    [arrayOfStrings addObject:[NSString stringWithFormat:@"Object %d",arrayOfStrings.count]];  
    [self updateDisplay];  
}
```

Console Area:  
NSLog will print here. You can also  
See stacktraces when your program  
crashes.  
**Can use gdb commands here as well!**

Buggy > Thread 1 > 0 -[BuggyViewController updateDisplay]

Local	All Output
<ul style="list-style-type: none"><li>self = (BuggyViewController *) 0x4d39540</li><li>_cmd = (struct objc_selector *) 0x2ce4</li></ul>	<pre>GNU gdb 6.3.50-20050815 (Apple version gdb-1518) (Sat Feb 12 02:52:12 UTC 2011) 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".sharedlibrary apply-load-rules a ll Attaching to process 14344. Pending breakpoint 1 - ""BuggyViewController.m":26" resolved Current language: auto; currently objective-c (gdb)</pre>

The image shows a screenshot of the Xcode IDE with a GDB debugging session. The top pane displays Objective-C code for `BuggyViewController`. The code includes an interface with a property `arrayOfStrings` and an implementation with a method `updateDisplay`. A callout box points to the `arrayOfStrings` variable in the variables window, stating: "Variables window. You can inspect variables by expanding the arrows".

```
//  
#import "BuggyViewController.h"  
  
@interface BuggyViewController()  
@property (readonly) NSMutableArray * arrayOfStrings;  
@end  
  
@implementation BuggyViewController  
  
@synthesize arrayOfStrings;  
  
-(void) updateDisplay  
{  
    arrayOfStrings  
}  
  
-(IBAction)addObject:(id)sender  
{  
    [arrayOfStrings addObject:[NSString stringWithFormat:@"Object %d", arrayOfStrings.count]];  
    [self updateDisplay];  
}
```

The bottom pane shows the GDB console with the following output:

```
GNU gdb 6.3.50-20050815 (Apple version gdb-1518) (Sat Feb 12 02:52:12 UTC 2011)  
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".sharedlibrary apply-load-rules a  
ll  
Attaching to process 14344.  
Pending breakpoint 1 - ""BuggyViewController.m":26" resolved  
Current language: auto; currently objective-c  
(gdb)
```

The variables window shows the following local variables:

```
Local  
A self = (BuggyViewController *) 0x4d39540  
A _cmd = (struct objc_selector *) 0x2ce4
```

The All Output pane shows the GDB console output. The status bar at the bottom indicates "Thread 1: Stopped after step".

```

@implementation BuggyViewController

@synthesize arrayOfStrings, arrayContentsLabel;

-(void) updateDisplay
{
    arrayContentsLabel.text = [arrayOfStrings description];
}

-(IBAction) addObject:(id) sender
{
    [arrayOfStrings addString:[NSString stringWithFormat:@"Object %d", arrayOfStrings.count]];
    [self updateDisplay];
}

-(IBAction) removeObject:(id) sender
{
    [arrayOfStrings removeLastObject];
    [self updateDisplay];
}

-(void) dealloc

```

Mouse over a local variable and you can see more information about it

[arrayOfStrings] NSMutableArray \* arrayOfStrings 0x0 0 objects Thread 1: Stopped at breakpoint 1

```
//  
  
#import "BuggyViewController.h"  
  
@interface BuggyViewController()  
@property (readonly) NSMutableArray * arrayOfStrings;  
@end  
  
@implementation BuggyViewController  
  
@synthesize arrayOfStrings,arrayContentsLabel;  
  
-(void) updateDisplay  
{  
    arrayContentsLabel.text = [arrayOfStrings description];  
}  
  
-(IBAction)addObject:(id)sender  
{  
    [arrayOfStrings addString:[NSString stringWithFormat:@"Object %d",arrayOfStrings.count]];  
    [self updateDisplay];  
}
```

Set break points by clicking on the left near your code

No Assistant

Thread 1: Stopped after step

Buggy > Thread 1 > 0 -[BuggyViewController updateDisplay]

Local
self = (BuggyViewController *) 0x4d39540
_cmd = (struct objc_selector *) 0x2ce4

All Output

GNU gdb 6.3.50-20050815 (Apple version gdb-1518) (Sat Feb 12 02:52:12 UTC 2011)  
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".sharedlibrary apply-load-rules all  
Attaching to process 14344.  
Pending breakpoint 1 - ""BuggyViewController.m":26" resolved  
Current language: auto; currently objective-c  
(gdb)



Right Click the breakpoint to get more options and set break conditions

The screenshot shows the Xcode interface with a breakpoint set at line 26 of BuggyViewController.m. A context menu is open over the breakpoint, displaying the following options:

- BuggyViewController.m:26
- Condition
- Ignore  time(s) before stopping
- Action Click to add an action
- Options  Automatically continue after evaluating actions

A "Done" button is located at the bottom right of the dialog. The background code shows the `updateDisplay` method. The bottom of the screen shows the local variables pane with `self = (BuggyViewController *) 0x4d3...` and `UIViewController = (UIViewController) {...}`, and the console output for the GNU gdb 6.3.50-20050815.

