

iPhone App Dev

Lesson 5

Source Codes

<https://github.com/makzan/ios-dev-course-example-2013>

Exercise

- What app are you developing?
- Any ideas worth discussing?
- Any great app design worth sharing?

Summary

- Loading network image with AFNetworking
- Introducing Category
- Declaring Category methods
- Querying Facebook profile pic
- Accessing NSDictionary
- UIScrollView
- Arranging Elements with UIScrollView

Loading Network Image with AFNetworking

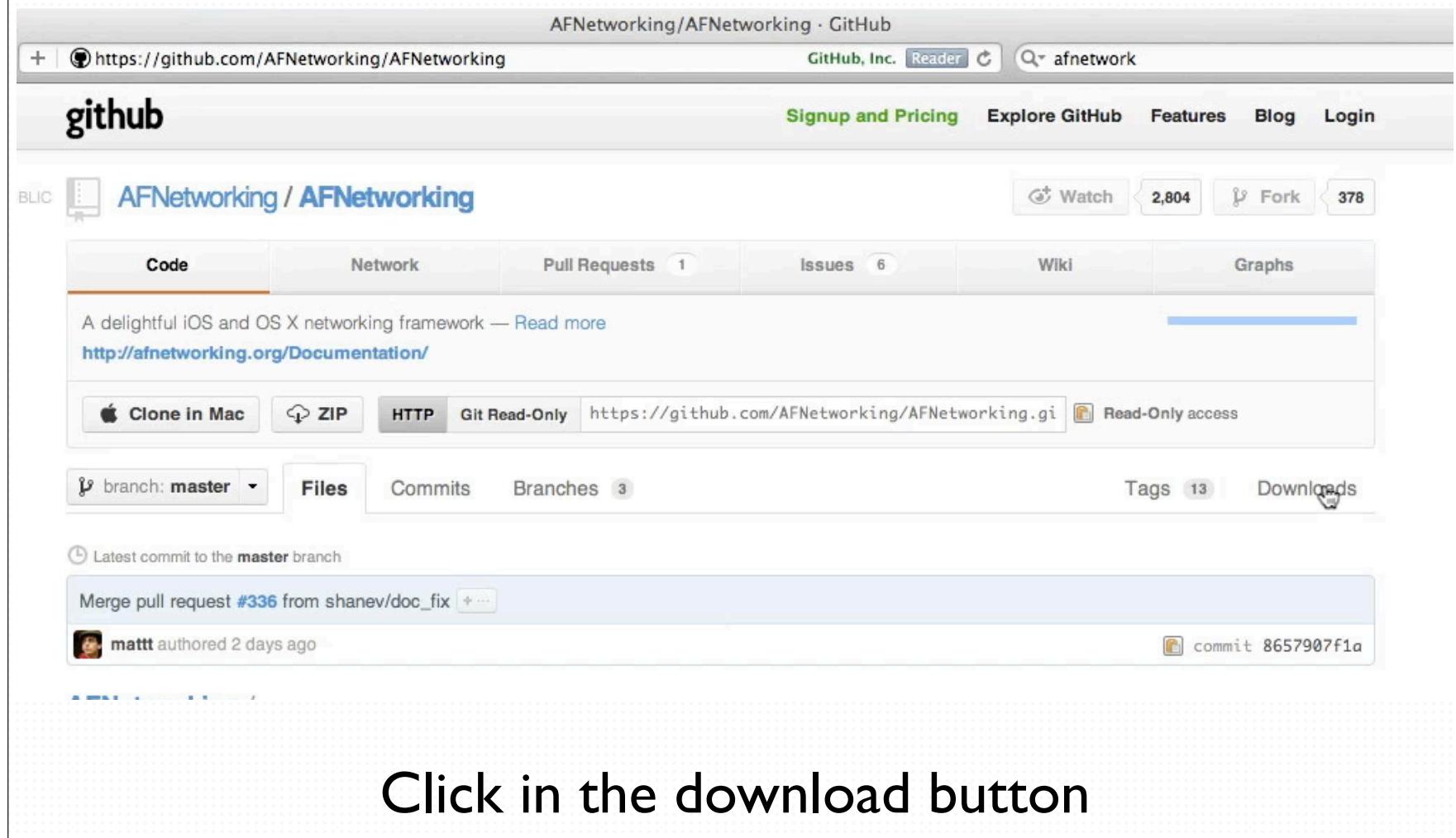
Including AFNetworking

A screenshot of a Google search results page for the query "afnetworking". The search bar at the top contains "afnetworking". The results page shows a "Search" section with the message "About 30,800 results (0.31 seconds)". On the left, there is a sidebar with links for "Everything", "Images", "Maps", "Videos", "News", "Shopping", and "More". Below these are "Show search tools" and "More search tools". The main content area displays several search results:

- AFNetworking/AFNetworking · GitHub**
https://github.com/AFNetworking/AFNetworking - Cached
1 day ago – A delightful iOS and OS X networking framework. Contribute to AFNetworking development by creating an account on GitHub.
»
- Getting Started with ...**
Getting Started with AFNetworking.
Page History. This is a guide to ...
- (AFNetworking) · GitHub**
AFNetworking Website. Last updated January 23, 2012 ...
- AFNetworking FAQ**
How is AFNetworking affected by Gowalla's acquisition ...
More results from github.com »
- Pull Requests 4**
Loading... Read; Open; Closed.
Submitted; Updated; Popularity ...
- Commits**
AFQueryStringComponentsFromKeyA always returns at ...
- AFOAuth2Client**
AFOAuth2Client - AFNetworking Extension for OAuth 2 ...

Search the **AFNetworking** code in github.

Including AFNetworking



A screenshot of a GitHub repository page for 'AFNetworking / AFNetworking'. The page title is 'AFNetworking / AFNetworking · GitHub'. The repository has 2,804 stars and 378 forks. The 'Code' tab is selected. Below the tabs, there's a brief description: 'A delightful iOS and OS X networking framework — [Read more](#)' and a link to the documentation: '<http://afnetworking.org/Documentation/>'. There are download links for 'Clone in Mac', 'ZIP', 'HTTP', and 'Git Read-Only'. The URL is <https://github.com/AFNetworking/AFNetworking.git>. A 'Read-Only access' button is also present. Below these, there are buttons for 'branch: master', 'Files', 'Commits', 'Branches (3)', 'Tags (13)', and 'Downloads'. A tooltip indicates the latest commit to the master branch. A merge pull request #336 from shanev/doc_fix is shown, authored by mattt 2 days ago, with a commit hash of 8657907f1a. At the bottom of the page, a large call-to-action text says 'Click in the download button'.

AFNetworking / AFNetworking · GitHub

github

Signup and Pricing Explore GitHub Features Blog Login

BLIC AFNetworking / AFNetworking Watch 2,804 Fork 378

Code Network Pull Requests 1 Issues 6 Wiki Graphs

A delightful iOS and OS X networking framework — [Read more](#)

<http://afnetworking.org/Documentation/>

Clone in Mac ZIP HTTP Git Read-Only <https://github.com/AFNetworking/AFNetworking.git> Read-Only access

branch: master Files Commits Branches (3) Tags (13) Downloads

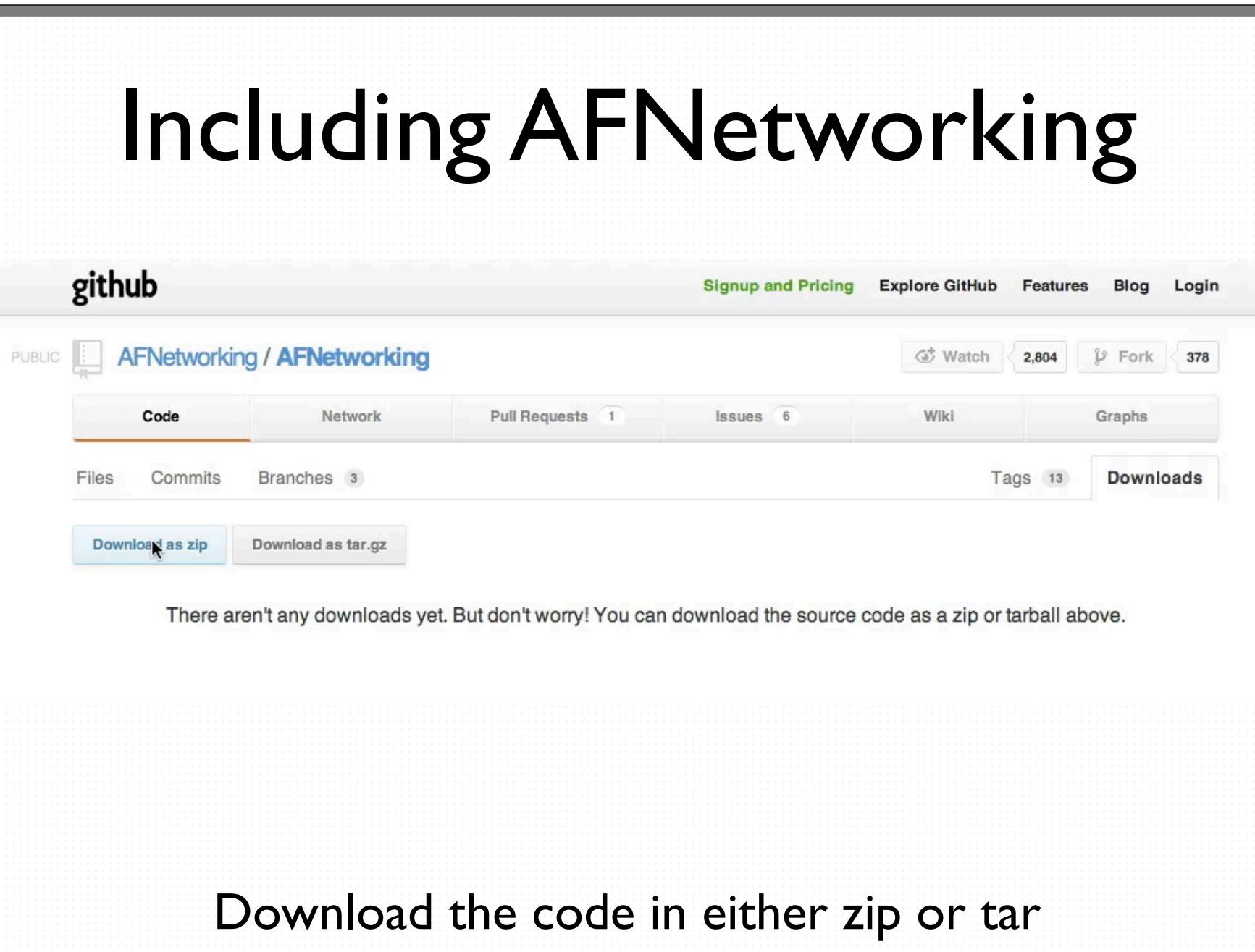
Latest commit to the **master** branch

Merge pull request #336 from shanev/doc_fix + ...

mattt authored 2 days ago commit 8657907f1a

Click in the download button

Including AFNetworking

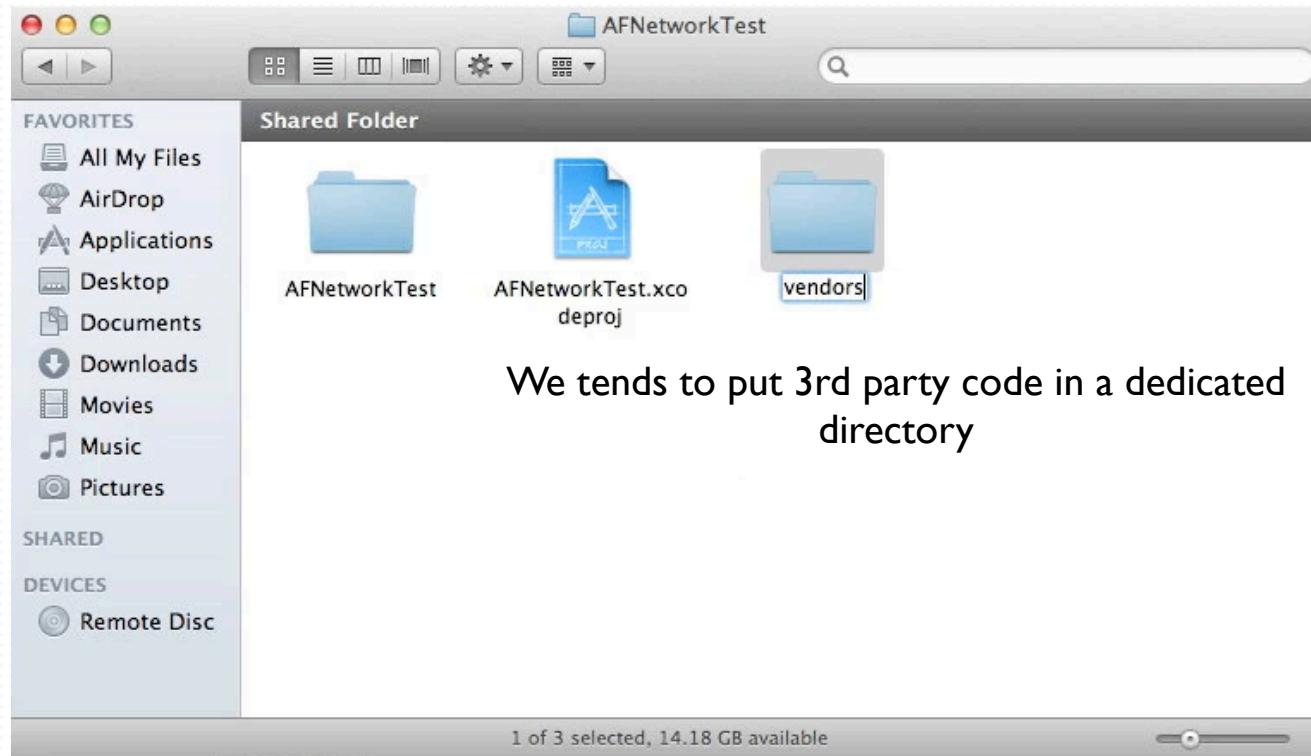


A screenshot of a GitHub repository page. The repository is named "AFNetworking / AFNetworking". The "Code" tab is selected. Below the tabs, there are links for "Files", "Commits", and "Branches" (with 3 branches). On the right, there are links for "Tags" (with 13 tags) and "Downloads". Two download buttons are visible: "Download as zip" and "Download as tar.gz". A cursor is hovering over the "Download as zip" button. The GitHub header includes links for "Signup and Pricing", "Explore GitHub", "Features", "Blog", and "Login".

There aren't any downloads yet. But don't worry! You can download the source code as a zip or tarball above.

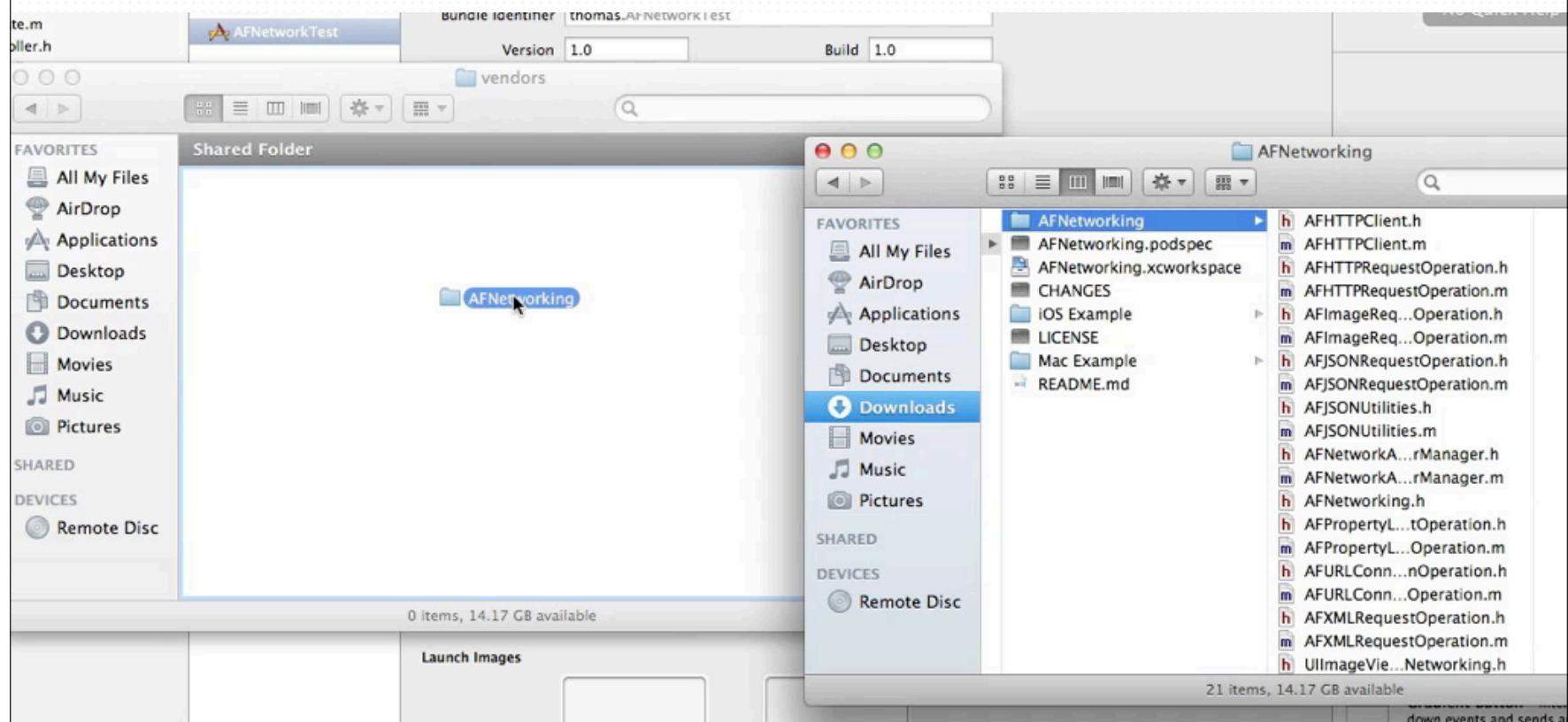
Download the code in either zip or tar

Including AFNetworking



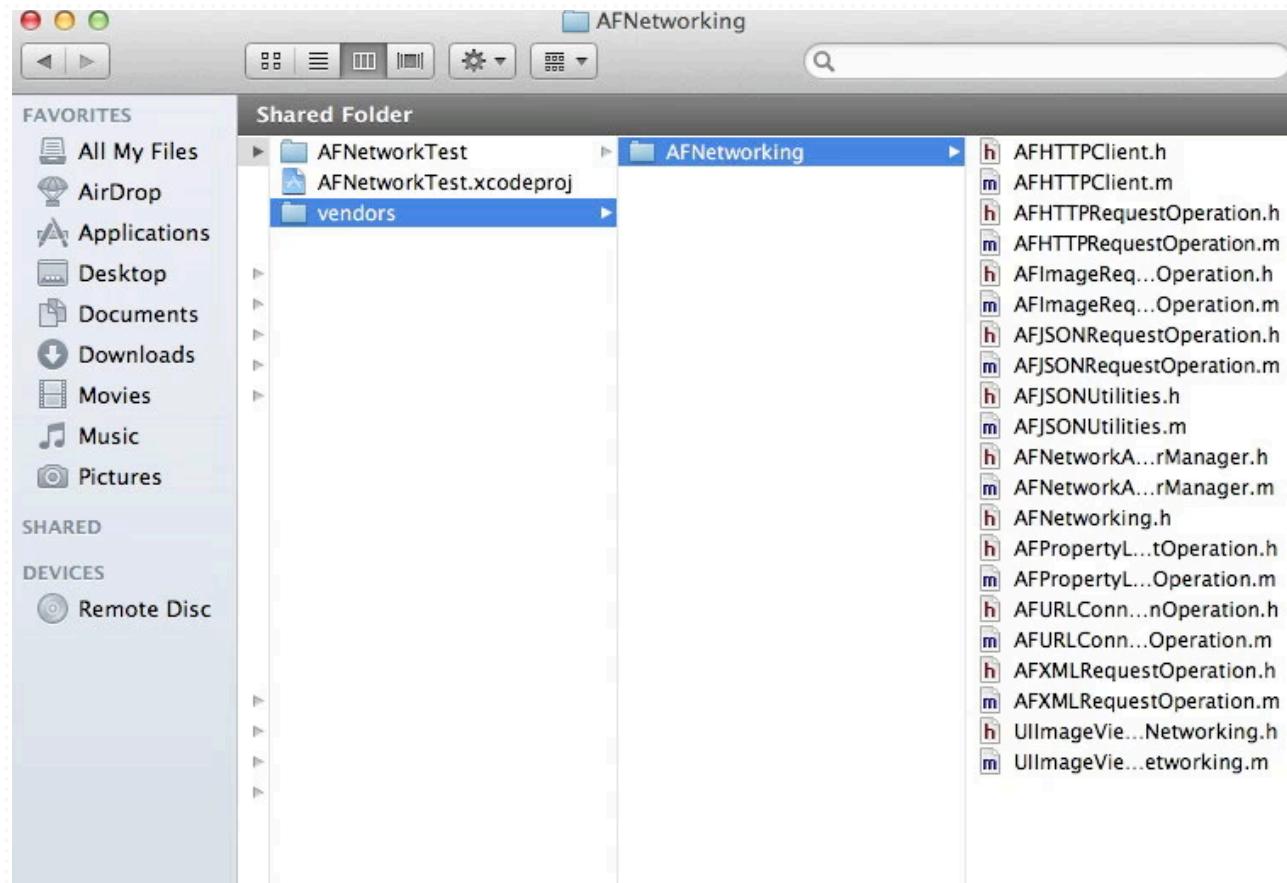
Prepare a folder named **vendors** for 3rd party code

Including AFNetworking



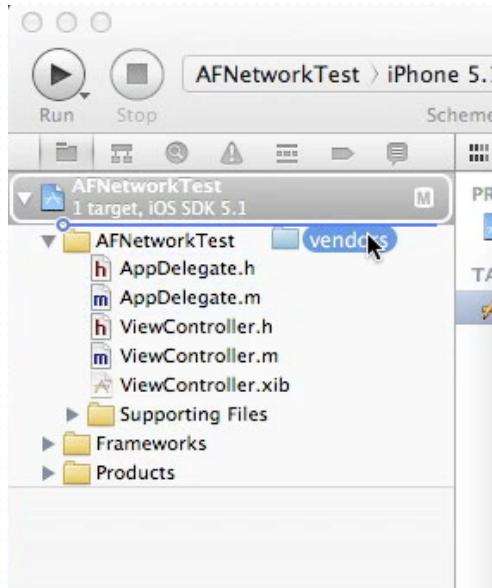
Put the AFNetworking source code into vendors

Including AFNetworking



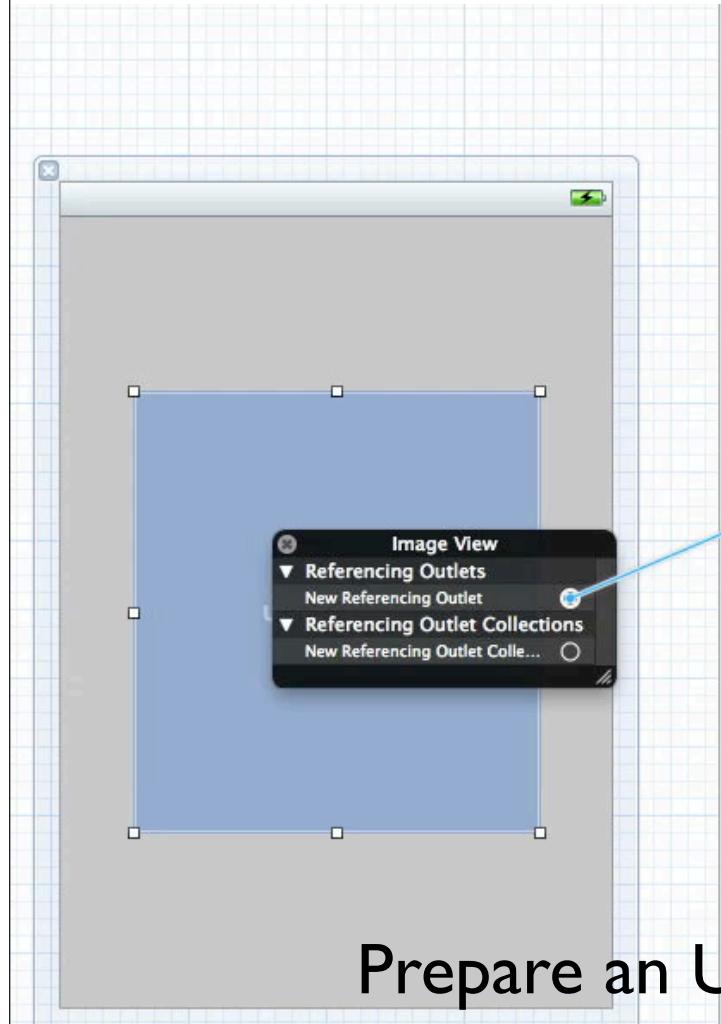
This is what our **vendors** folder looks like

Including AFNetworking



Add the vendors to XCode project tree

Loading Network Images



```
//  
// ViewController.h  
// AFNetworkTest  
//  
// Created by Freshman on 5/24/12.  
// Copyright (c) 2012 __MyCompanyName__. All rights  
reserved.  
  
//  
  
#import <UIKit/UIKit.h>  
  
@interface ViewController : UIViewController  
  
@end Insert Outlet
```

Prepare an UIImageView in interface

Loading Network Images

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

@interface ViewController : UIViewController
@property (retain, nonatomic) IBOutlet UIImageView *
    networkImageView;
@end
```

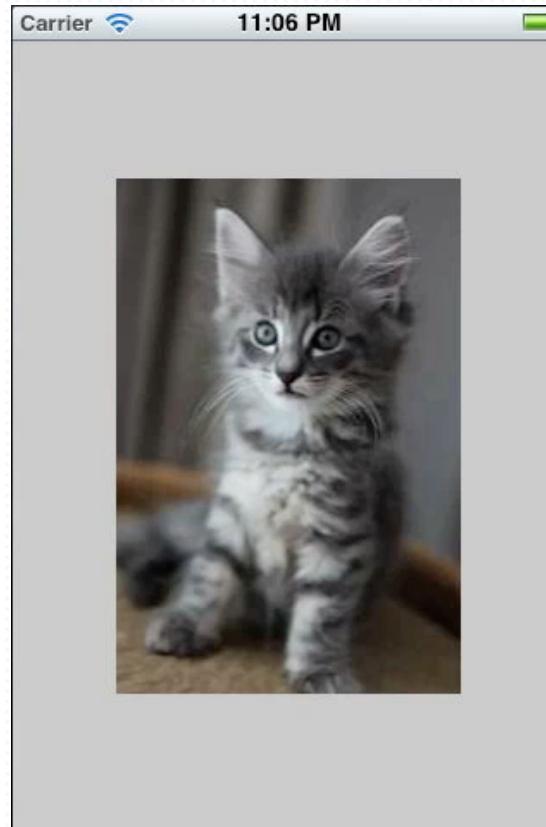
In this example, we named it `networkImageView`

Loading Network Images

```
#import "ViewController.h"  
  
#import "UIImageView+AFNetworking.h"  
  
@interface ViewController ()  
  
@end  
  
@implementation ViewController  
  
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a nib.  
  
    NSURL * imageURL = [NSURL URLWithString:@"http://placekitten.com/200/300"];  
    [self.networkImageView setImageWithURL:imageURL];  
}
```

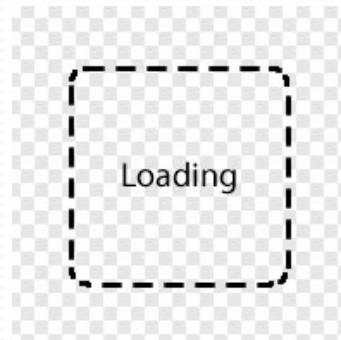
Import the UIImageView+Network.h and set the image URL

Loading Network Images



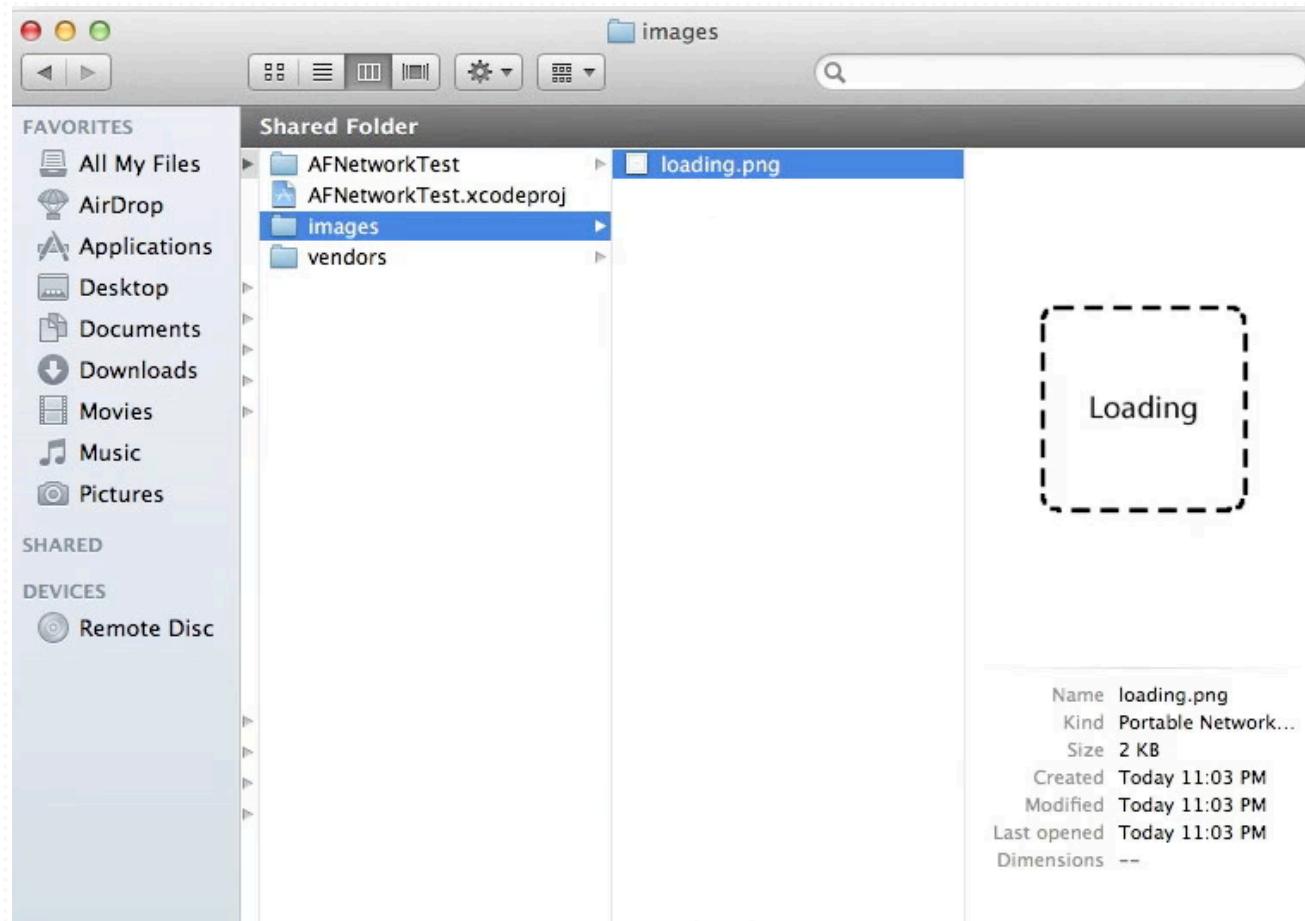
Run it in simulator and we get the image from network.

Loading Placeholder



Prepare an image to show before the target image is loaded

Loading Placeholder



Put the placeholder image into project

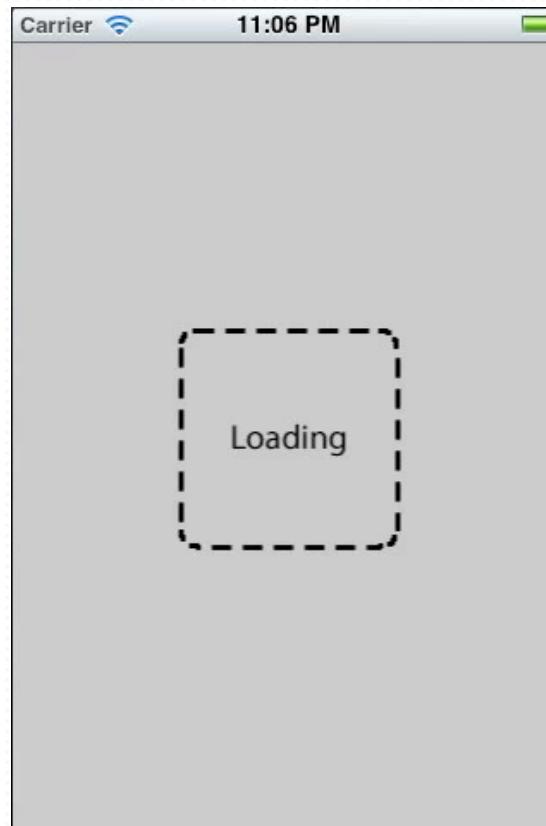
Loading Placeholder

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    NSURL * imageURL = [NSURL URLWithString:@"http://placekitten.com/200/300"];
    [self.networkImageView setImageWithURL:imageURL placeholderImage:[UIImage
        imageNamed:@"loading.png"]];
}
```

Modify the network image code with placeholder option

Loading Placeholder



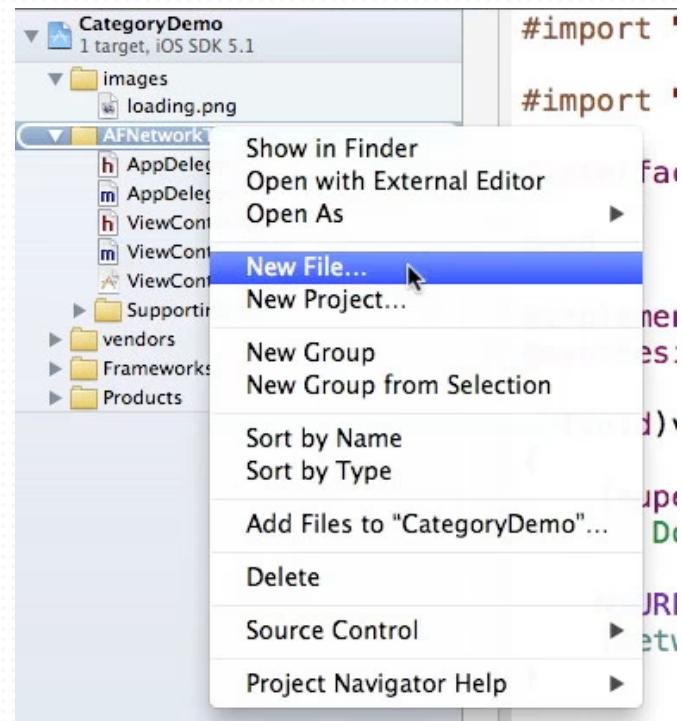
The placeholder show up before the image loaded in simulator.

Introducing Category

What is Category

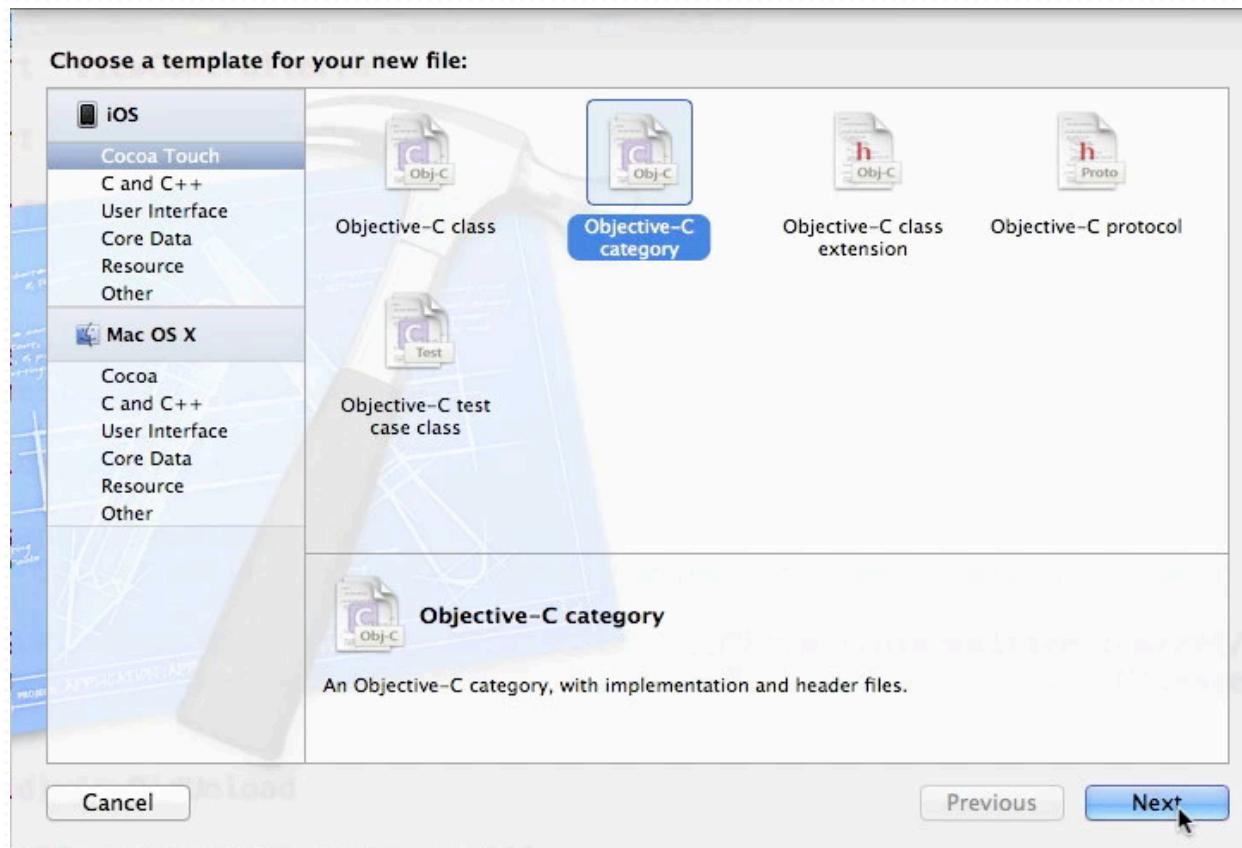
- It adds methods behaviors to Class at Run Time
- It is usually named in following format:
UIView+Position
- and the interface and implementation is
@interface UIView (Position)

Category Example



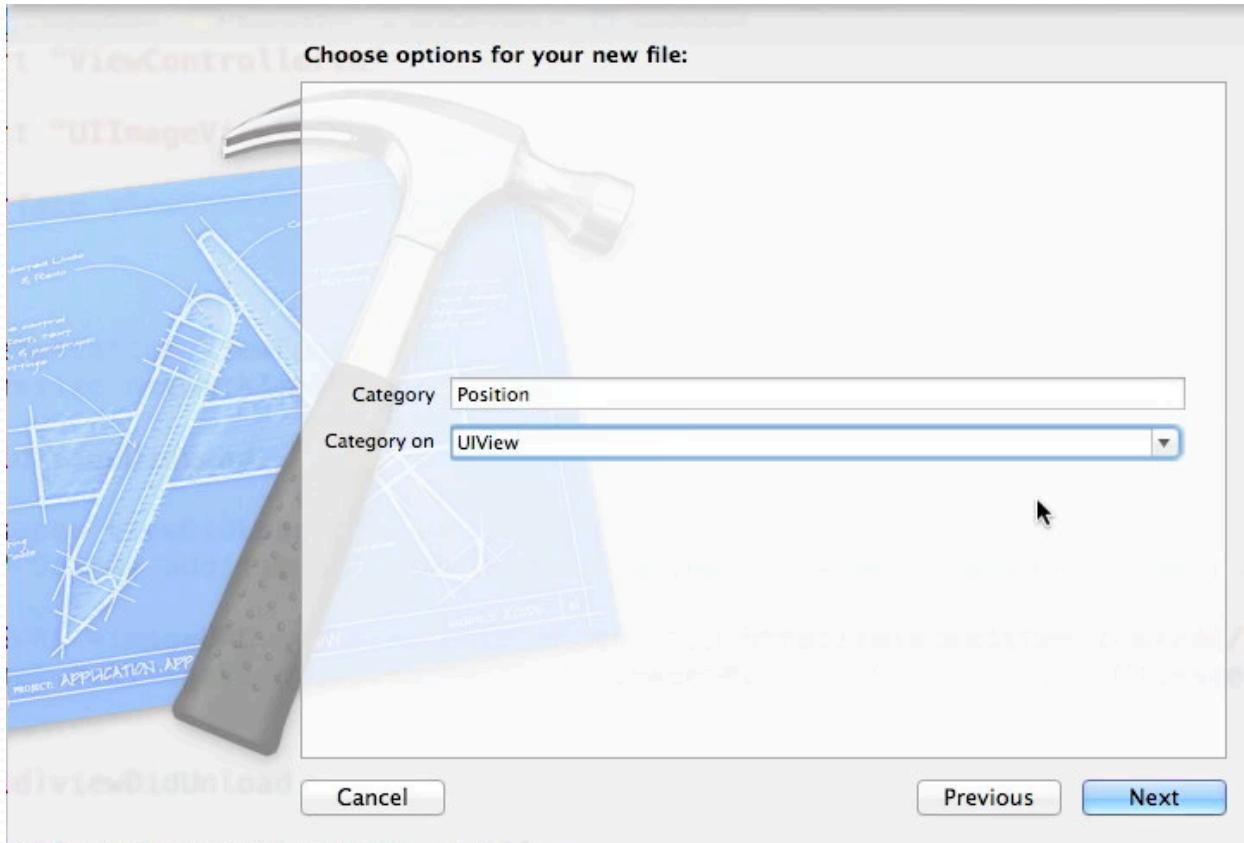
Create a new file in project

Category Example



Choose Category file

Category Example



Name the category Position on class UIView

Category Example

```
//  
// UIView+Position.m  
// CategoryDemo  
//  
// Created by Freshman on 5/24/12.  
// Copyright (c) 2012 __MyCompanyName__. All  
rights reserved.  
  
//  
  
#import "UIView+Position.h"  
  
@implementation UIView (Position)  
  
- (void)setX:(float)x andY:(float)y  
{  
    CGRect frame = self.frame;  
    frame.origin.x = x;  
    frame.origin.y = y;  
    self.frame = frame;  
}  
  
@end
```

```
//  
// UIView+Position.h  
// CategoryDemo  
//  
// Created by Freshman on 5/24/12.  
// Copyright (c) 2012 __MyCompanyName__. All  
rights reserved.  
  
//  
  
#import <UIKit/UIKit.h>  
  
@interface UIView (Position)  
  
// set the position of the UIView to (x,y)  
- (void)setX:(float)x andY:(float)y;  
  
@end
```

Put above code in the category files

Category Example

```
#import "UIImageView+AFNetworking.h"  
#import "UIView+Position.h"
```

Back to ViewController class and import our category

Category Example

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    NSURL * imageURL = [NSURL URLWithString:@"http://placekitten.com/200/300"];
    [self.networkImageView setImageWithURL:imageURL placeholderImage:[UIImage
        imageNamed:@"loading.png"]];

    [self.networkImageView setX:0 andY:0];
}
```

Add our defined UIView position method

Category Example

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    NSURL * imageURL = [NSURL URLWithString:@"http://placekitten.com/200/300"];
    [self.networkImageView setImageWithURL:imageURL placeholderImage:[UIImage
        imageNamed:@"loading.png"]];

    [self.networkImageView setX:0 andY:0];
    [UIView animateWithDuration:5 animations:^{
        [self.networkImageView setX:320 andY:480];
    }];
}
```

It's boring. Let's animate it with animate method

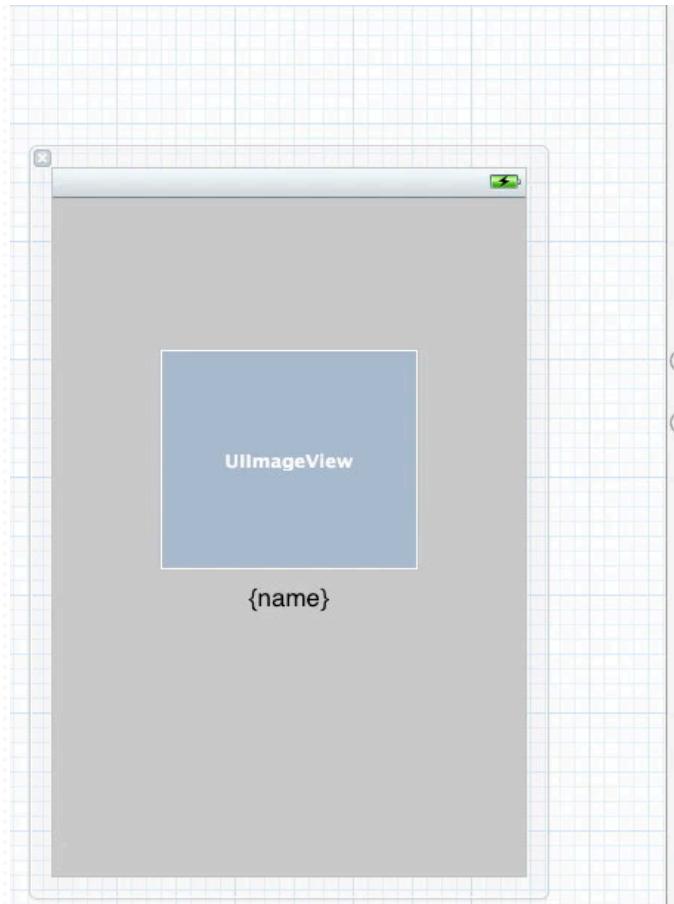
Category Example



The result with the cat moving from top left to bottom right

Querying Facebook Profile Picture

Loading FB Profile Pic



```
//  
//  ViewController.h  
//  AFNetworkTest  
//  
//  Created by Freshman on 5/24/12.  
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved  
  
#import <UIKit/UIKit.h>  
  
@interface ViewController : UIViewController  
@property (retain, nonatomic) IBOutlet UIImageView *  
    networkImageView;  
@property (retain, nonatomic) IBOutlet UILabel *nameLabel;  
@end
```

Add a label in interface

Loading FB Profile Pic

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    NSURL * imageURL = [NSURL URLWithString:@"http://graph.facebook.com/makzan/picture?type=large"];
    [self.networkImageView setImageWithURL:imageURL placeholderImage:[UIImage imageNamed:@"loading.png"]];

}
```

Change the image URL to your profile pic. Done.

Facebook Graph API

Loading JSON

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    NSURL * imageURL = [NSURL URLWithString:@"http://graph.facebook.com/makzan/picture?type=large"];
    [self.networkImageView setImageWithURL:imageURL placeholderImage:[UIImage imageNamed:@"loading.png"]];

    NSURL * url = [NSURL URLWithString:@"http://graph.facebook.com/makzan"];
    NSURLRequest * request = [NSURLRequest requestWithURL:url];

    AFJSONRequestOperation * operation = [AFJSONRequestOperation JSONRequestOperationWithRequest:request
                                         success:^(NSURLRequest * request, NSHTTPURLResponse * response, id JSON) {
        NSLog(@"Result: %@", JSON);
        self.nameLabel.text = [JSON objectForKey:@"name"];
    } failure:^(NSURLRequest * request, NSHTTPURLResponse * response, NSError * error, id JSON) {
        // Handle error here.
    }];
    [operation start];
}
```

We use the AFJSONRequestOperation to query API in JSON format.

Accessing NSDictionary

Accessing NSDictionary

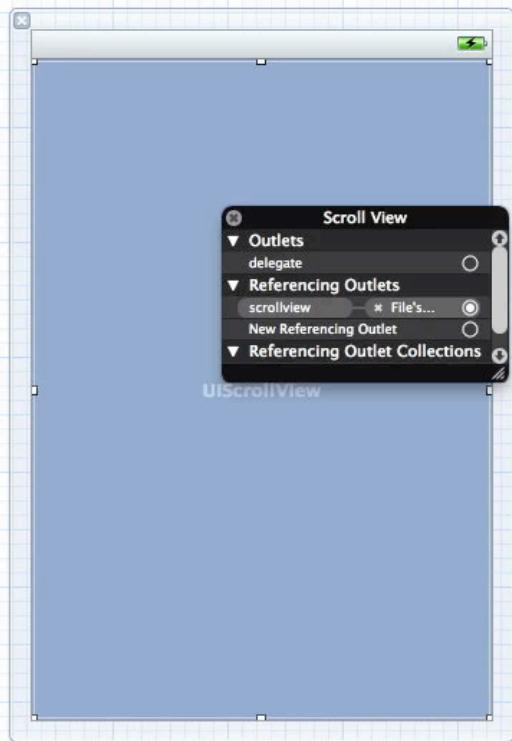
```
// dict is a NSDictionary instance.  
[dict objectForKey:@"key"];
```

Accessing NSDictionary

```
// dict is a NSDictionary instance.  
[dict setValue:@"John" forKey:@"name"];
```

Using UIScrollView

UIScrollView



Drag a UIScrollView into view and put it full screen.
Name it scrollview

UIScrollView

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    UIImageView *image = [[UIImageView alloc] initWithImage:[UIImage imageNamed:@"320x460.png"]];
    [self.scrollview addSubview:image];

    [self.scrollview setContentSize:CGSizeMake(320*4, 460)];
}
```

We add an image view inside scroll view

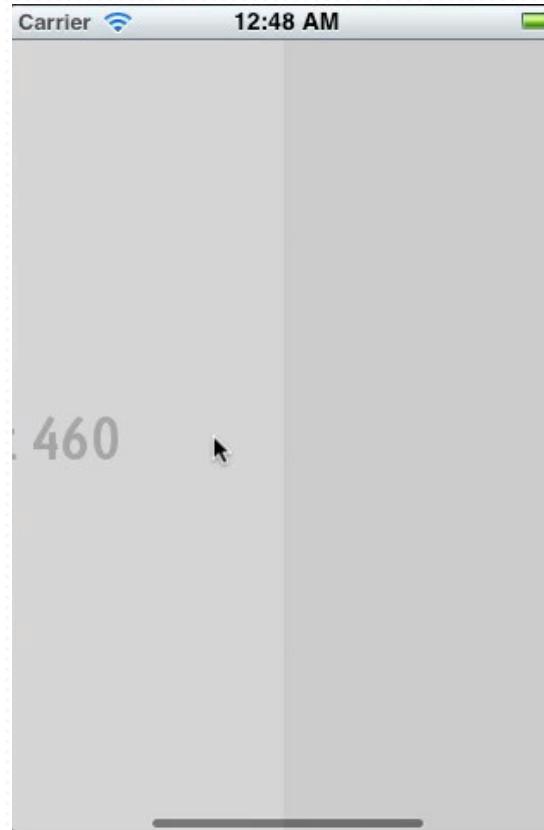
Subview

- Every UIView contains sub views.
- Sub views are also UIView.
- When we move the parent views, all sub views follows.
- You can think UIView is a container of other UIViews.

Adding and Removing Subview

- [view addSubview: view2];
- [view2 removeFromSuperview];
- [view insertSubview:view2 atIndex:2];
- [view insertSubview:view2 aboveSubview: view3];
- [view insertSubview:view2 belowSubview: view3];

UIScrollView



Test it in iPhone and we get the image inside scroll view

UIScrollView

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    for (int i=0; i<4; i++)
    {
        UIImageView *image = [[UIImageView alloc] initWithImage:[UIImage imageNamed:@"320x460.png"]];
        [self.scrollview addSubview:image];

        [image setX:320 * i andY:0];
    }

    [self.scrollview setContentSize:CGSizeMake(320*4, 460)];
}
```

Let's try add 4 images inside the scroll view

UIScrollView

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    for (int i=0; i<4; i++)
    {
        UIImageView *image = [[UIImageView alloc] initWithImage:[UIImage imageNamed:@"320x460.png"]];
        [self.scrollView addSubview:image];

        [image setX:320 * i andY:0];
    }

    [self.scrollView setContentSize:CGSizeMake(320*4, 460)];
    self.scrollView.pagingEnabled = YES;
}
```

We can enable page by page scrolling with pagingEnabled

Using Interface Builder with UIScrollView

UIScrollView in Interface Builder

- Easy to accidentally put views inside UIScrollView
- Need to arrange views carefully

Exercise

Combining all we learned so far

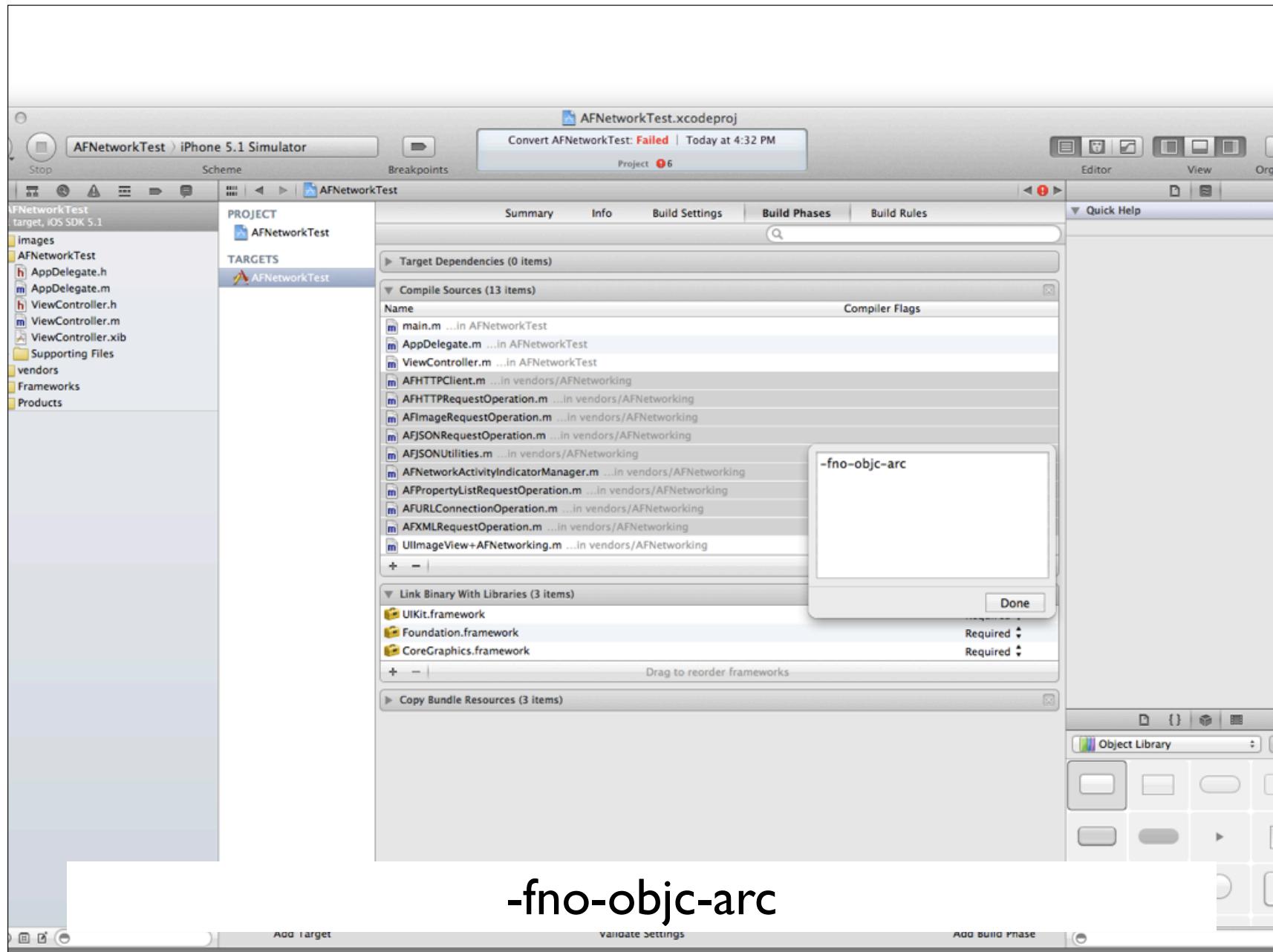


Query Facebook Photos API and display them in scroll view

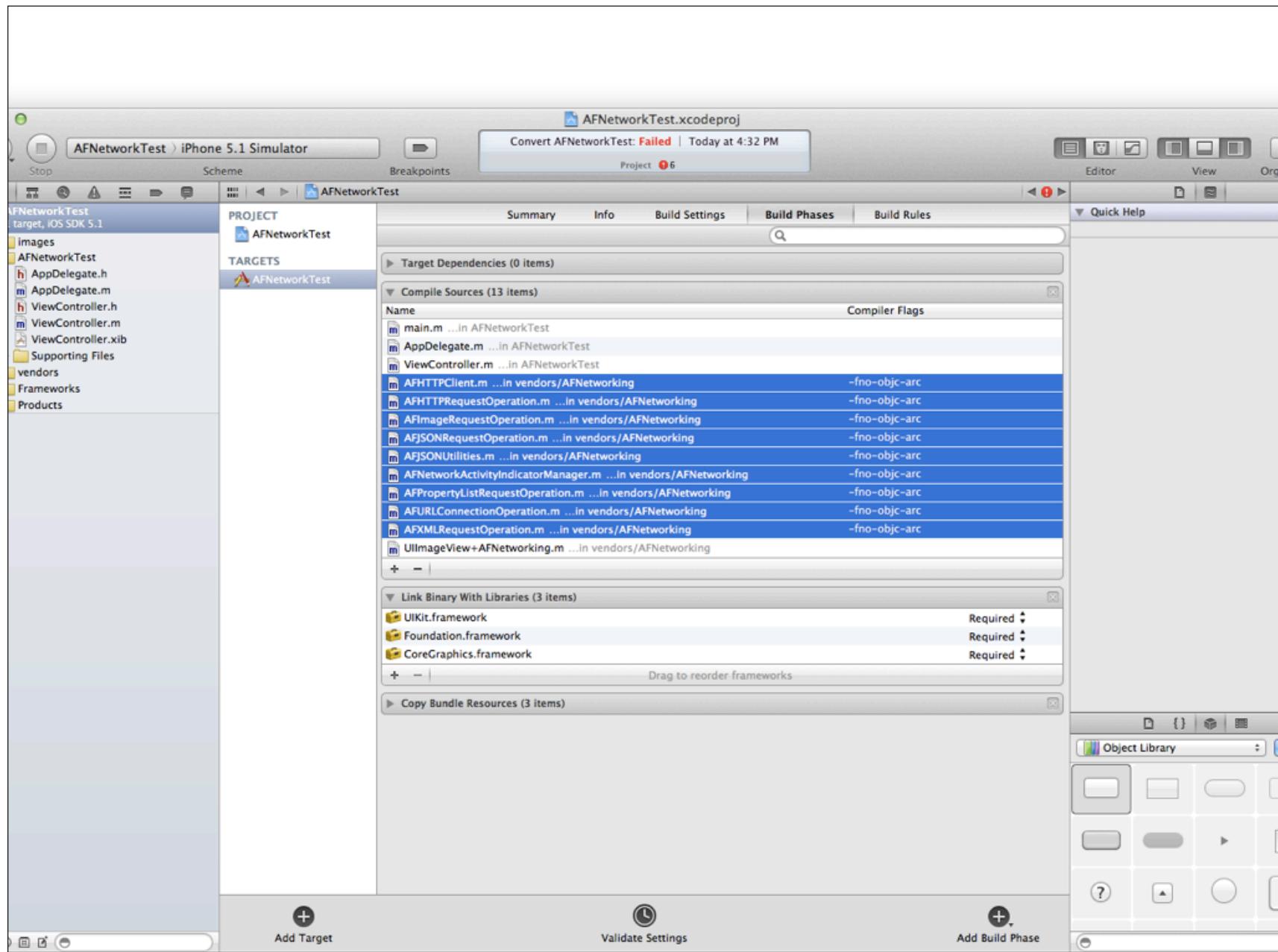
Avoiding files compiling with ARC

No ARC Flag

1. Project Setting
2. Build Phrase
3. Compile Sources
4. Apply -fno-objc-arc



-fno-objc-arc



4