

iOS Lab for Devs

Developing Apps for iPhone and iPad



Lesson 4

Topics of the day

Mattina

- Storyboard
- UICollectionView

Pomeriggio

- Debugging Techniques
- Publish Flow

Riassunto Frameworks iOS

Cocoa Touch

Foundation



Address Book UI
Game Kit
Map Kit
Message UI
Core Audio
Core Graphic
Core Animation
...

UIKit



Storyboard

DAPizzeria > iPhone 5.0 Simulator

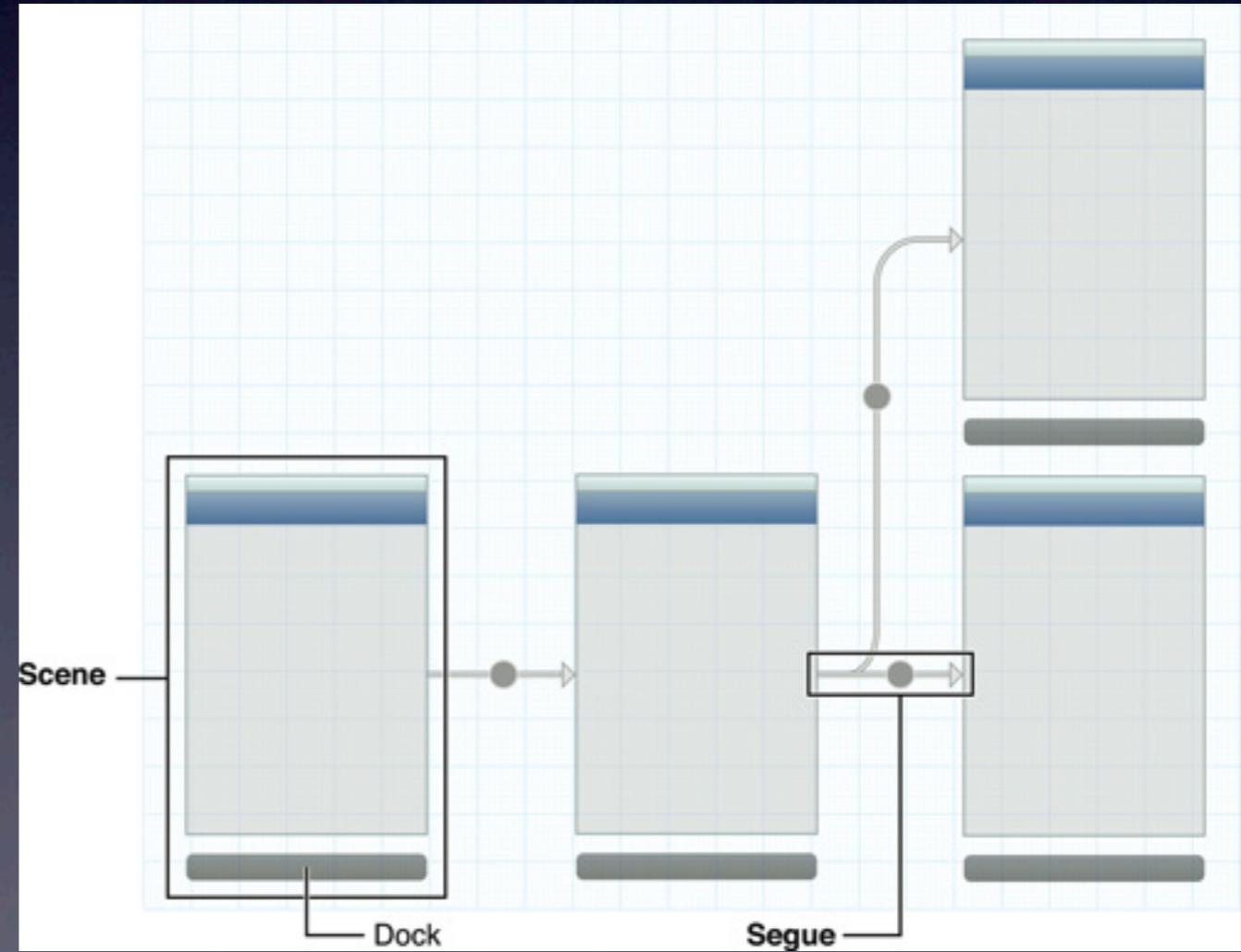
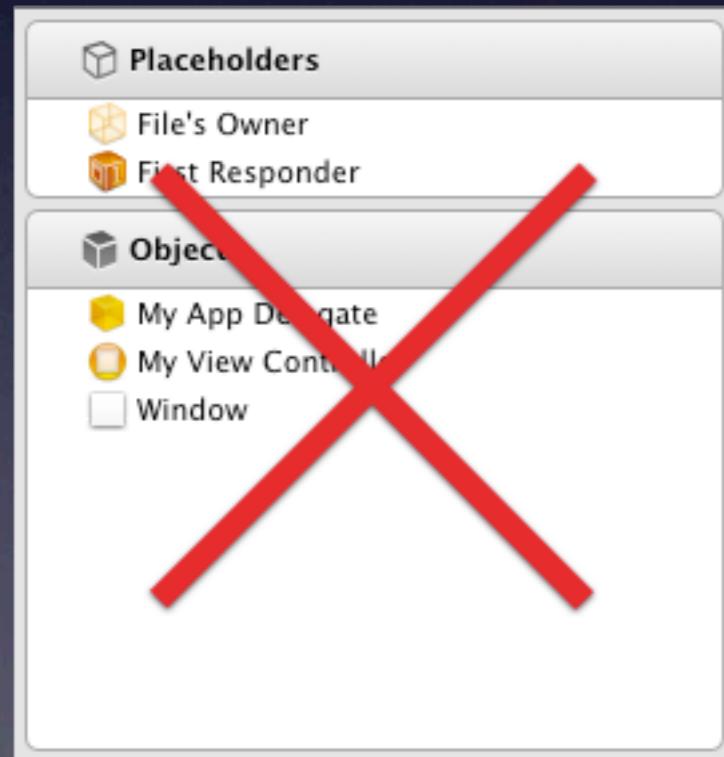
Analyze Succeeded | 19/07/12 at 21.50

```
45 UITabBarItem *menuTabBarItem = [[[UITabBarItem alloc] initWithTitle:@"Menu" image:menuImage tabBarItem];
46 menuViewController.tabBarItem = menuTabBarItem;
47
48 // Creo il navigation controller per il navigator
49
50 NSURLRequest *browserUrlRequest = [NSURLRequest requestWithURL:[NSURL URLWithString:kStartingURL]];
51
52 DABrowserViewController *browserViewController = [[[DABrowserViewController alloc] initWithNibName:@"DABrowserViewController" bundle:nil urlRequest:browserUrlRequest] autorelease];
53
54 UINavigationViewController *browserNavController = [[[UINavigationController alloc] initWithRootViewController:browserViewController] autorelease];
55 UIImage *browserImage = [UIImage imageNamed:@"rss"];
56
57 UITabBarItem *browserTabBarItem = [[[UITabBarItem alloc] initWithTitle:@"iOS Lab" image:browserImage tabBarItem];
58 browserNavController.tabBarItem = browserTabBarItem;
59
60 // Creo il navigation controller per il frigo
61 DAFridgeViewController *fridgeViewController = [[[DAFridgeViewController alloc] initWithNibName:@"DAFridgeViewController" bundle:nil] autorelease];
62
63 UINavigationViewController *fridgeNavController = [[[UINavigationController alloc] initWithRootViewController:fridgeViewController] autorelease];
64 UIImage *fridgeImage = [UIImage imageNamed:@"trolley"];
65
66 UITabBarItem *fridgeTabBarItem = [[[UITabBarItem alloc] initWithTitle:@"Fridge" image:fridgeImage tabBarItem];
67 fridgeNavController.tabBarItem = fridgeTabBarItem;
68
69 // Creo il controller per la cassa
70 DACounterViewController *counterViewController = [[[DACounterViewController alloc] initWithNibName:@"DACounterViewController" bundle:nil] autorelease];
71 UIImage *counterImage = [UIImage imageNamed:@"piggy_bank"];
72 UITabBarItem *counterTabBarItem = [[[UITabBarItem alloc] initWithTitle:@"Counter" image:counterImage tabBarItem];
73 counterViewController.tabBarItem = counterTabBarItem;
74
75 // Creo il controller per la mappa
76 DAMapViewController *mapViewController = [[[DAMapViewController alloc] initWithNibName:@"DAMapViewController" bundle:nil] autorelease];
77 UIImage *mapImage = [UIImage imageNamed:@"globe"];
78 UITabBarItem *mapTabBarItem = [[[UITabBarItem alloc] initWithTitle:@"Map" image:mapImage tabBarItem];
79 mapViewController.tabBarItem = mapTabBarItem;
```

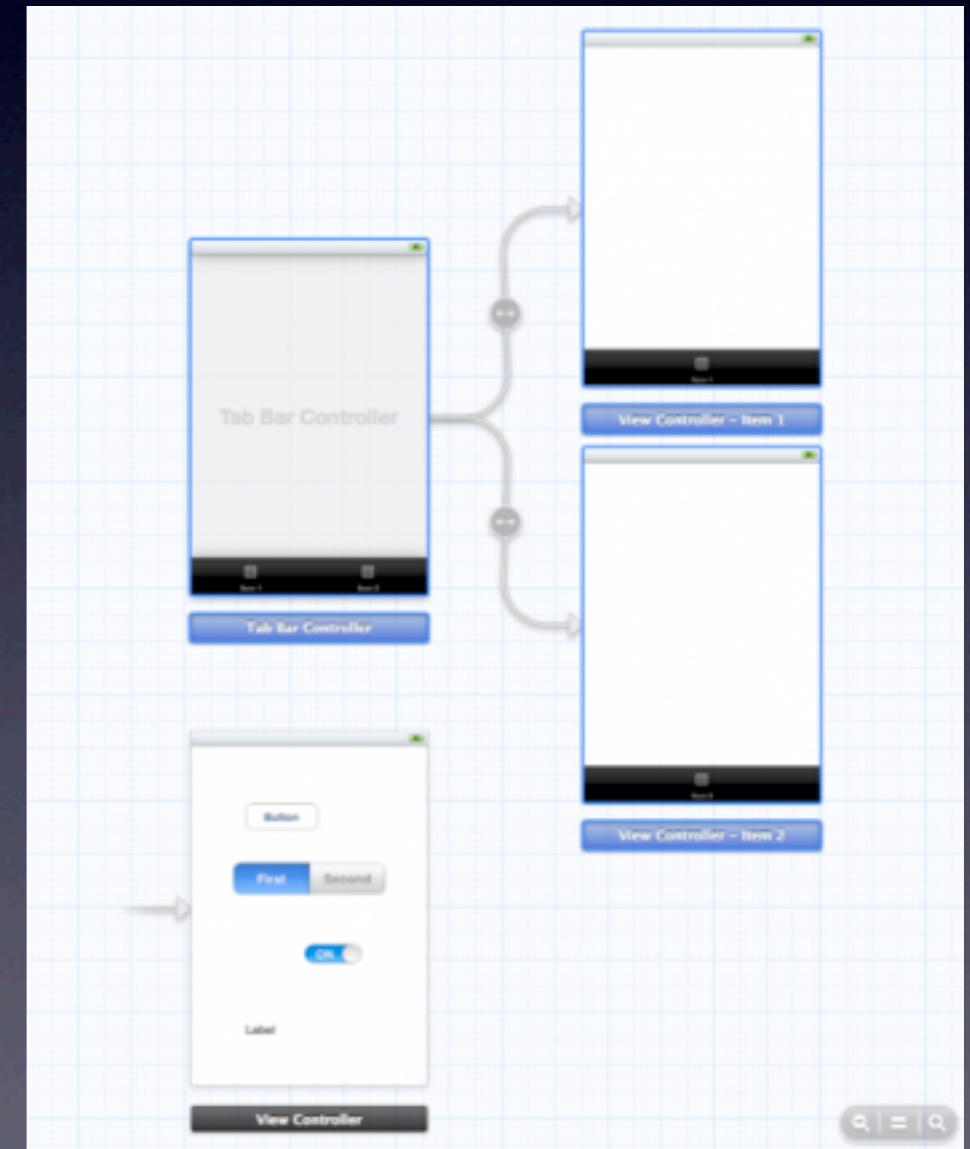
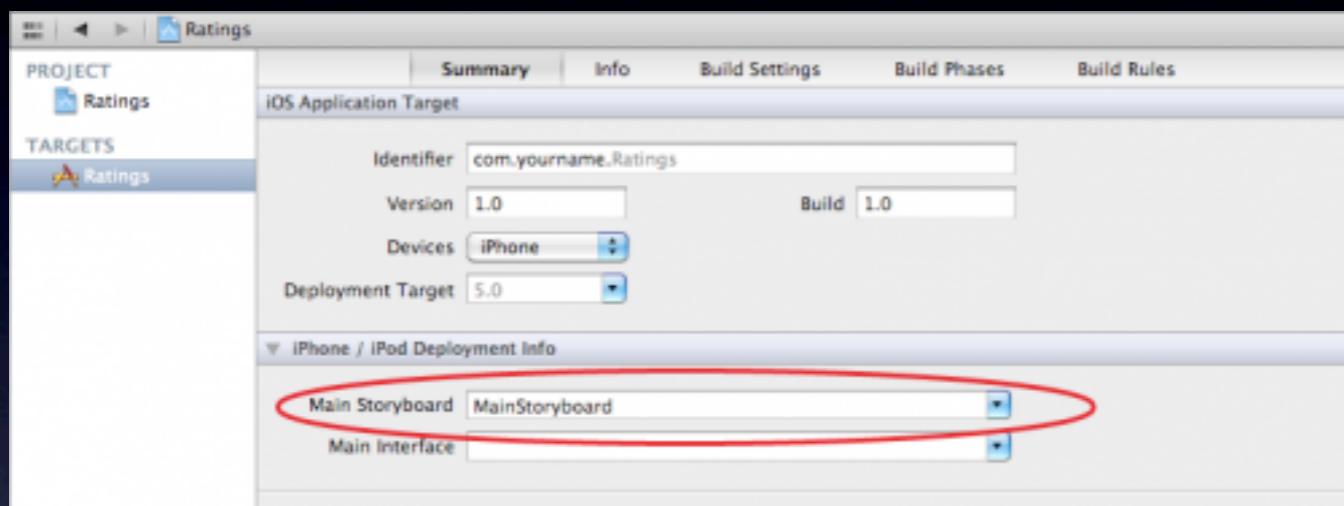


Storyboard

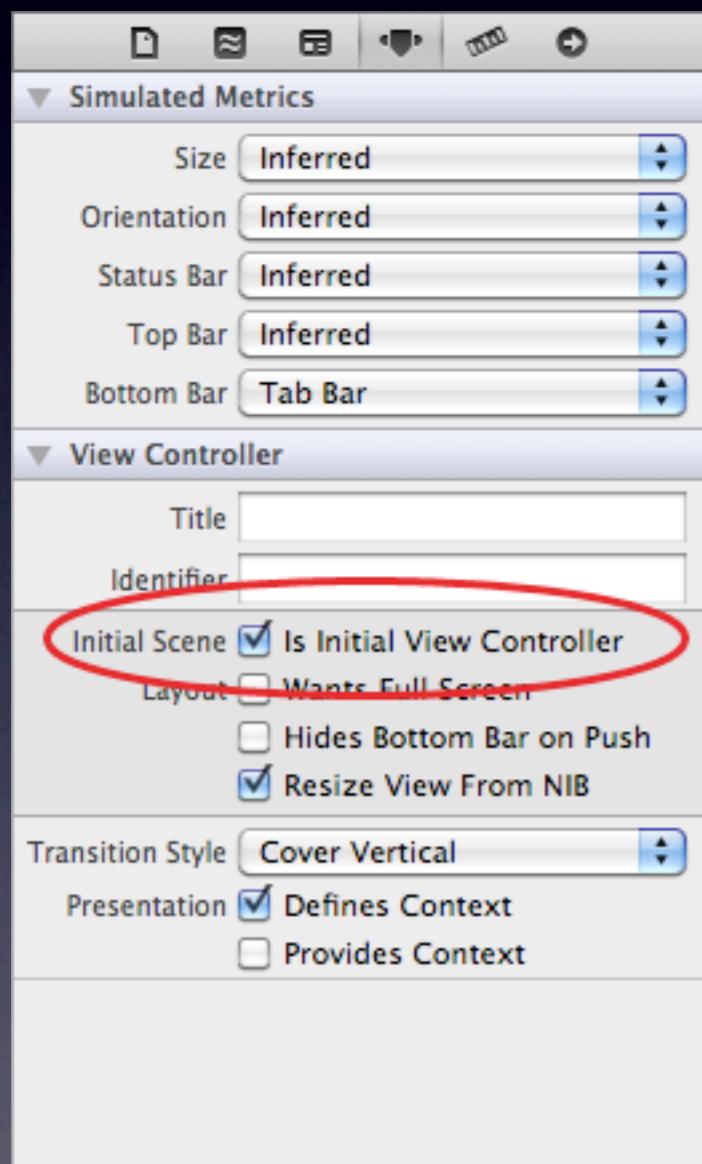
- Unico file .storyboard al posto di n file .xib
- Flusso utente gestito con le segue



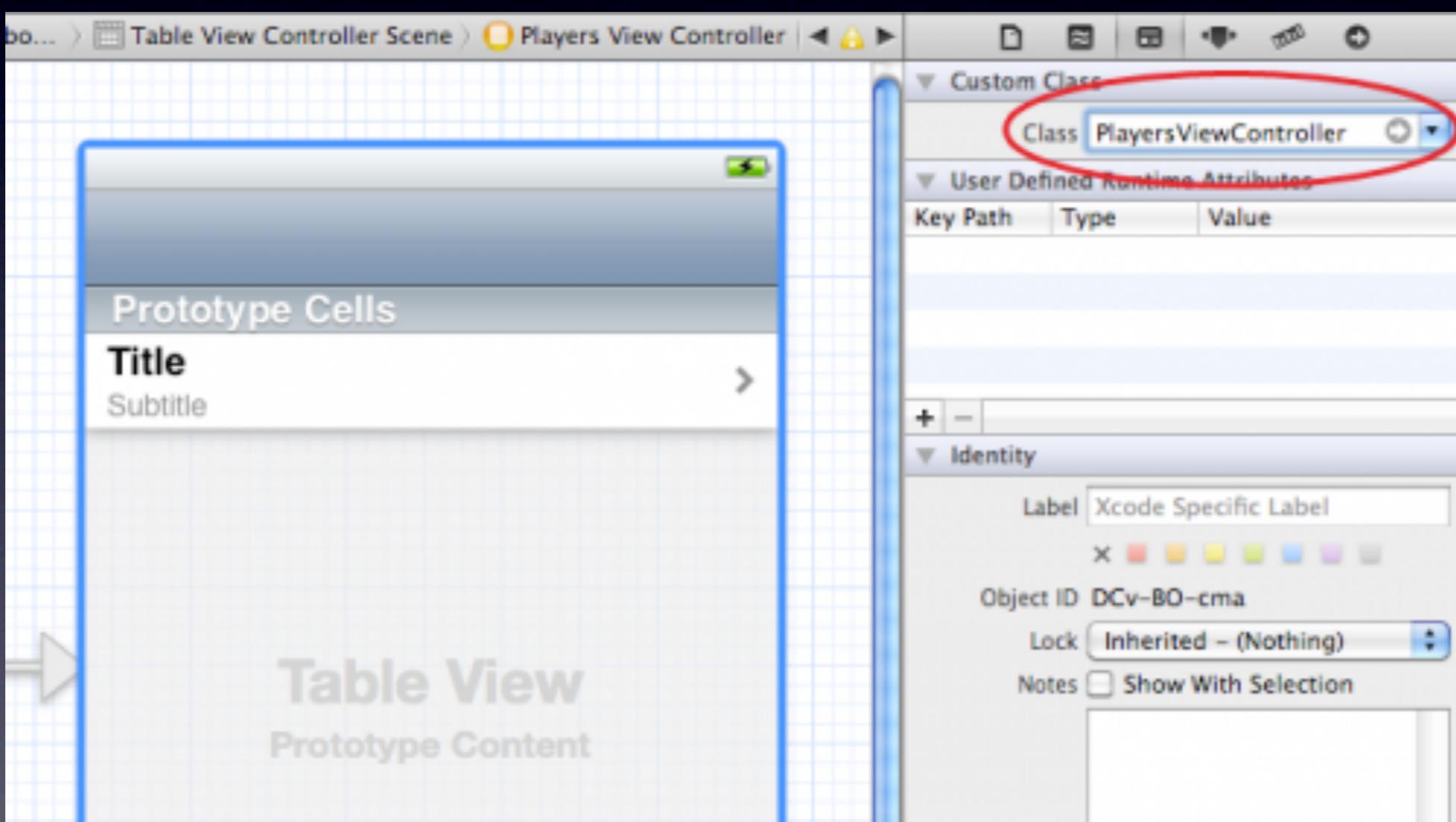
Storyboard



Storyboard



Storyboard

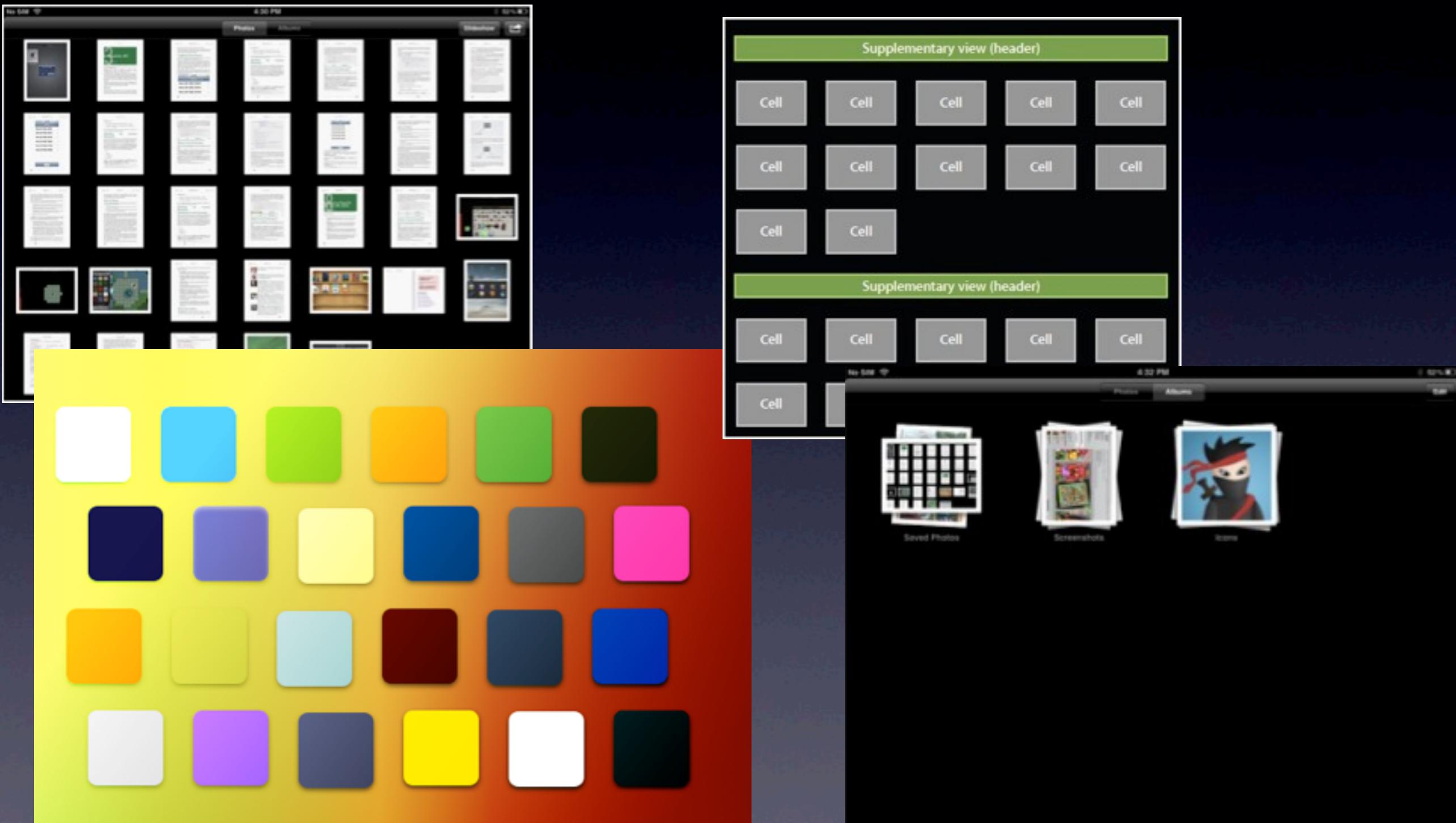


Metodi da conoscere

- `(void)prepareForSegue:(UIStoryboardSegue *)segue
sender:(id)sender`
- `(void)performSegueWithIdentifier:(NSString *)segueId
sender:(id)sender;`

UICollectionView

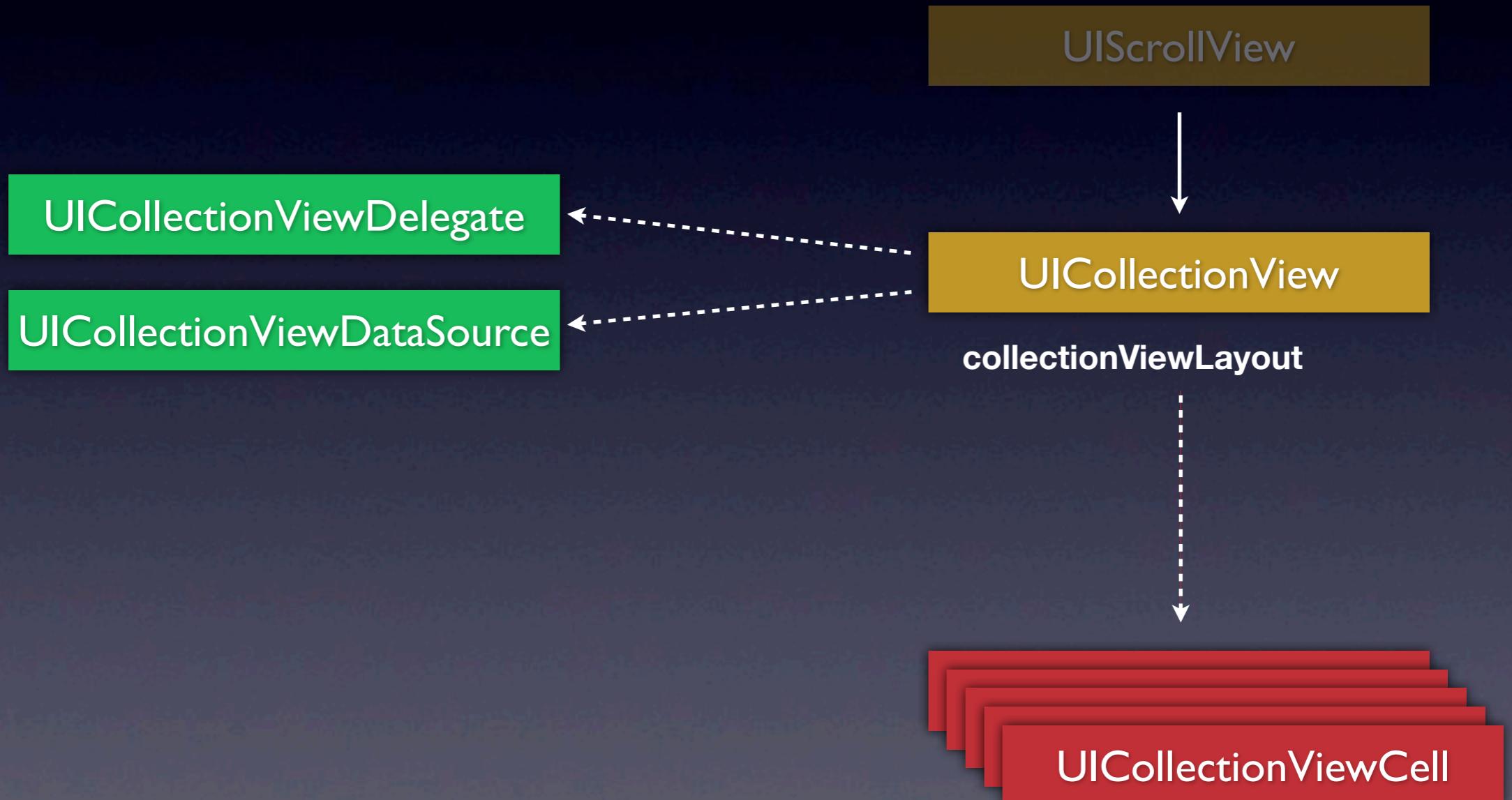
UICollectionView



UICollectionView



UICollectionView



Debugging techniques and performance tuning

LLVM e GCC

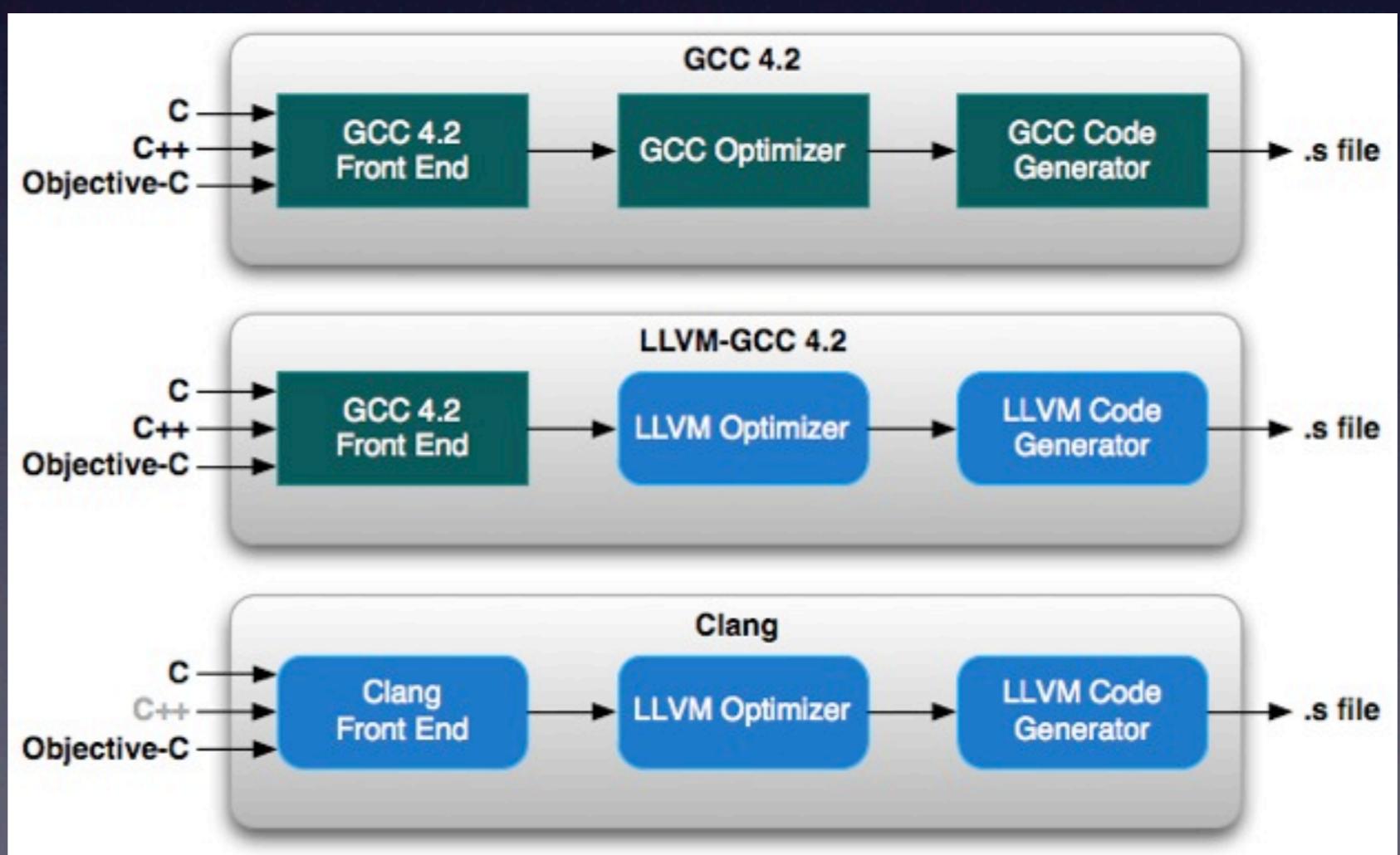
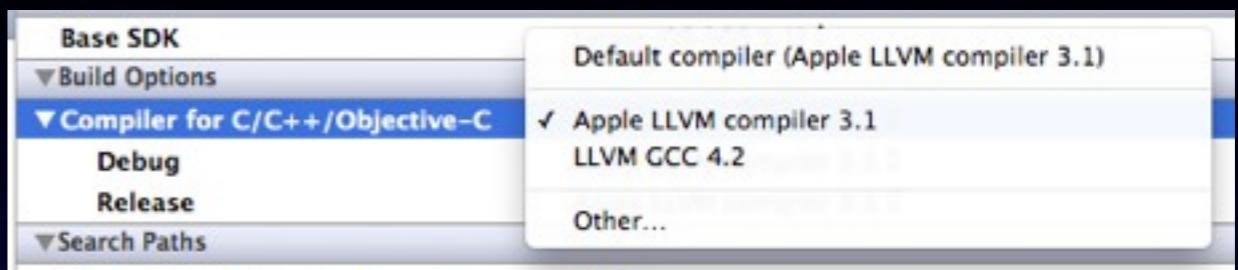
Premessa:

LLVM 3.0 è il compilatore di nuova generazione adottato da Apple.

E' il default ed è obbligatorio in Xcode 4.2.

GCC deprecato.

Reference: llvm.org/



Debugging techniques and performance tuning

Vedremo concetti riguardanti:

- Breakpoints (`objc_exception_throw`)
- Stack trace
- Leaks
- Heap
- Memory Warnings
- Messaggi di errore comuni
- Zombies objects
- SIGABRT and EXC_BAD_ACCESS exceptions
- treats warnings as errors (direttiva di compilazione)
- Instruments
- CLANG static analyzer

Breakpoints

Breakpoint condizionali

```
68 - (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath {  
69  
70     static NSString *CellIdentifier = @"LabelCell";  
71     NSUInteger row = [indexPath row];  
72  
73     UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];  
74     if (cell == nil) {  
75
```

Breakpoint	Location	Condition
-tableView:cellForRowAtIndexPath: - Line 73	NewViewController.m	row==50

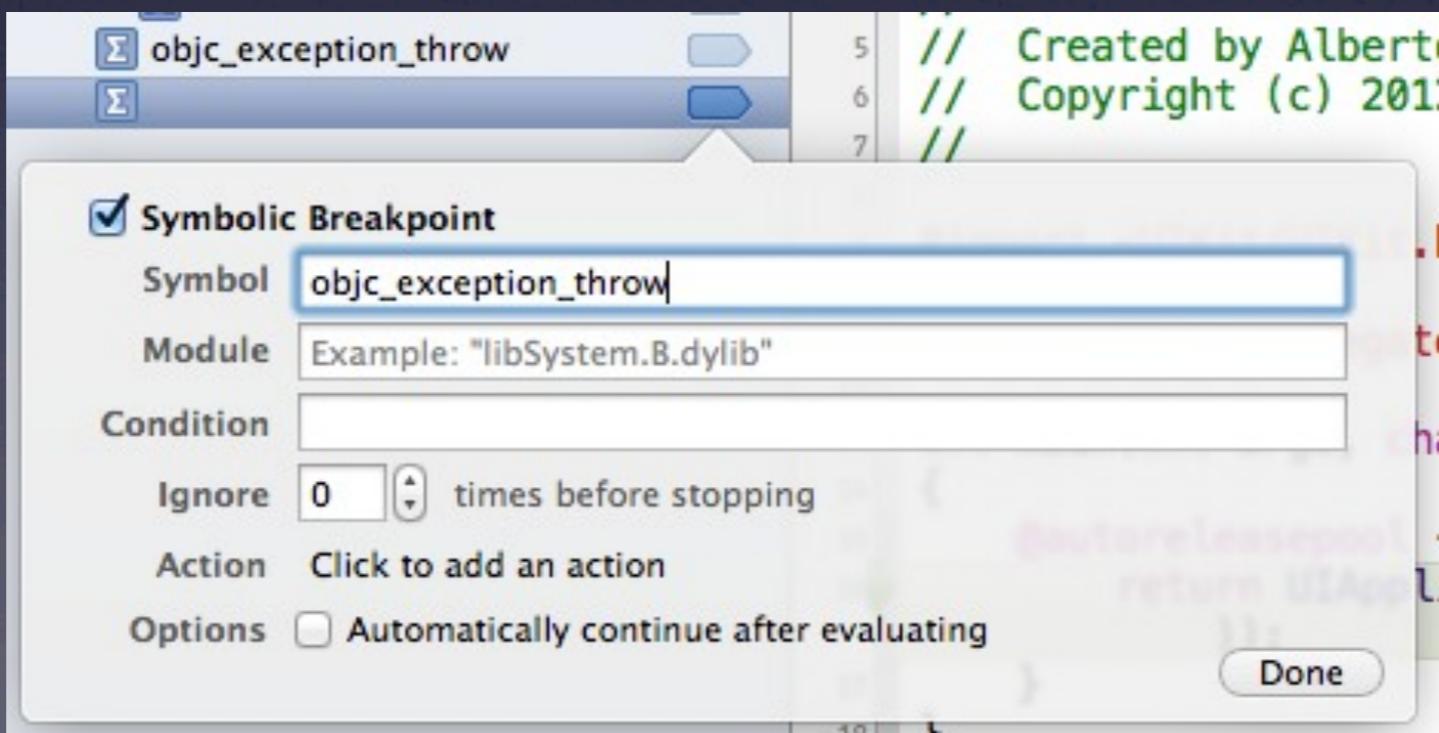
Breakpoint	Location	Condition
-tableView:cellForRowAtIndexPath: - Line 71	NewViewController.m	(NSUInteger)[indexPath row] == 50

objc_exception_throw breakpoint

In Objective-C non si usa spesso il try-catch e di solito le eccezioni arrivano al main (senza mai essere catturate).

In tal caso (e senza il breakpoint objc_exception_throw abilitato) il debugging è difficile e non si riesce a risalire allo statement scatenante l'eccezione.

objc_exception_throw blocca l'esecuzione al momento del lancio dell'eccezione.

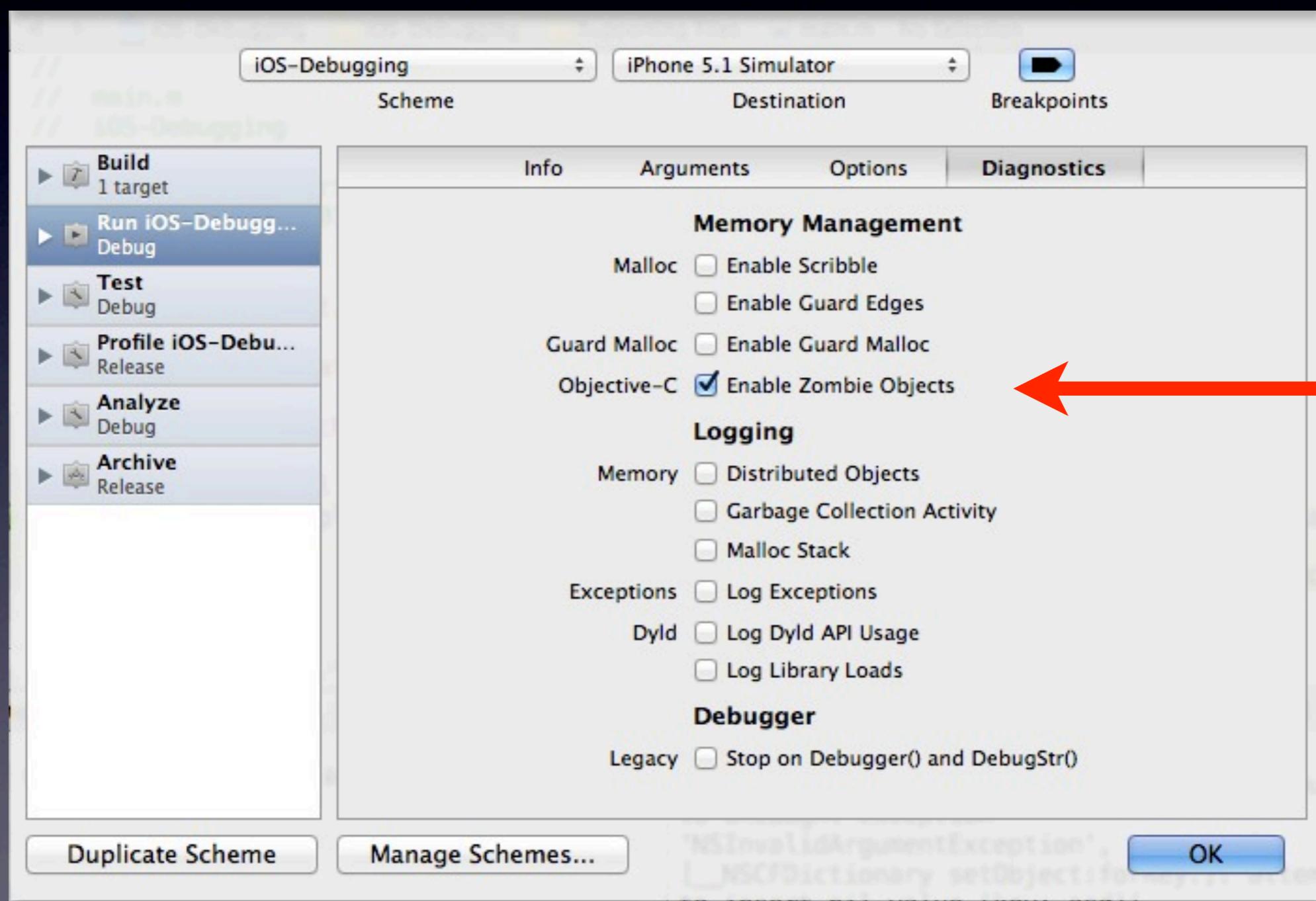


Enable Zombie Objects 1/2

Quando un oggetto viene deallocated, viene cancellato dalla memoria. In particolare con l'uso del messaggio autorelease, l'oggetto viene rilasciato (e deallocated) in un tempo successivo a noi sconosciuto. Se successivamente tentiamo di accedere all'oggetto, l'applicazione crasha a causa di un dangling pointer.

In tal caso è estremamente comodo abilitare la modalità “oggetti zombie”: quando un oggetto deve venire deallocated in realtà esso viene marcato come “zombie” ma rimane in memoria. Al successivo accesso (errato) possiamo dunque avere informazioni in console su che tipo di oggetto era stato referenziato al posto di avere un log pressoché inutile quale ‘message sent to deallocated instance’ che non ci farebbe (neanche lontanamente) sapere su quale oggetto stiamo gestendo la memoria in maniera errata.

Enable Zombie Objects 2/2



CLANG static Analyzer

Rileva leak dovuti a una gestione scorretta di retain e release.

```
NSObject *objectID = 0;

for (NSUInteger i=0; i < count; ++i) {      ▶ Looping back to the head of the loop

    NSObject *object = [trackedElements objectAtIndex:i];

    if ([object isKindOfClass:[NSString class]])
    {
        objectID = [[[NSString alloc] initWithString:aString];
        ▶ Method returns an Objective-C object with a +1 retain count (owning reference)
    }

    if (objectID != nil)
    {
        [objectID release];          ▶ Object released
    }                                ▶ Reference-counted object is used after it is released
}
```

CLANG static Analyzer

The screenshot shows the Xcode interface with the CLANG static Analyzer results window open. The left pane displays a tree view of issues found in various files across different projects. The right pane shows the source code for a file named `Model/ManagedObject.h`, with specific lines highlighted by a blue oval and a red exclamation mark, indicating potential errors or warnings.

Analyzer Results:

- NXLib project**: 1 issue
- NXLib.xcodeproj**: Validate Project Settings
- NXLib**: 7 issues
 - `UIDevice+IdentifierAddition.m`: Memory Error, Logic error
 - `UIImage+ProportionalFill.m`: Logic error
 - `MGSplitViewController.m`: Logic error, Dead store, Dead store, Logic error
 - `BWQuincyManager.m`: Memory (Core Foundation/Objective-C)
- dev1**: 16 issues
 - `NSUserDefaults+Additions.m`: Dead store, Dead store
 - `NXMagazinePageView.m`: Dead store
 - `NXCDVersion+Additions.m`: Dead store, Dead store
 - `NXCDWishlist+Additions.m`: Dead store
 - `NXNumberBadgeView.m`: Dead store
 - `NXSalesTableViewController.m`: Logic error
 - `CSSTokenizer.m`: Logic error, Logic error, Logic error
 - `NXCDUILabel+Additions.m`: Coding conventions (Apple), Potential null dereference. According to coding standards in 'Creating and Returning NSError Objects' the parameter " may be null
 - `NXWebViewHTMLOffline.m`: Coding conventions (Apple), Apple Mach-O Linker Error, Dsymutil Error

Code Editor (Model/ManagedObject.h):

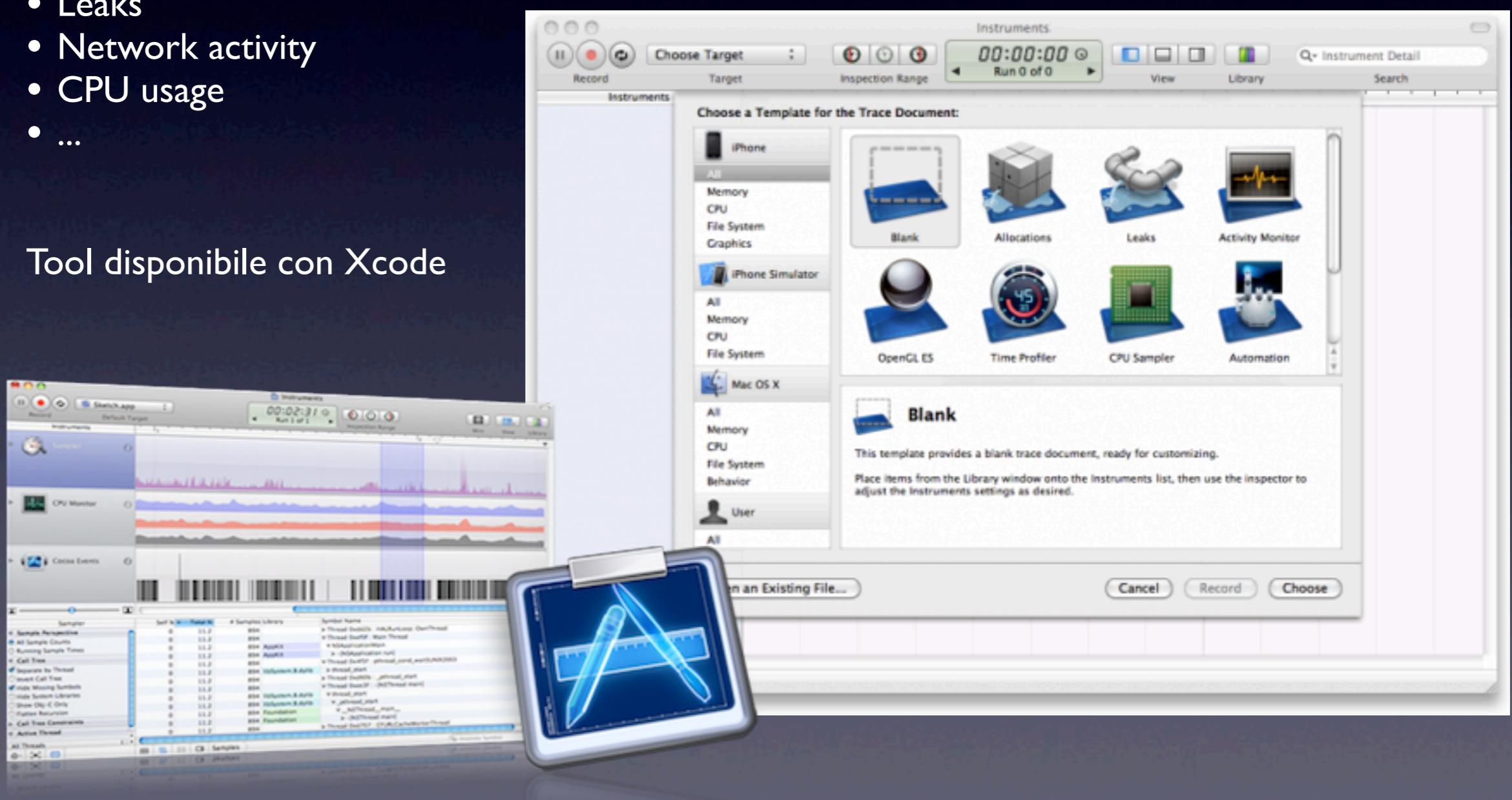
```
1. Potential null dereference. According to coding standards in 'Creating and Returning NSError Objects' the parameter " may be null
166     return kNXLBNotAssignedAgentCode;
167 }
168
169 + (NSString *)labelForOrderNote:(NSUInteger)index
170 {
171     return [LABEL stringAtIndex:index fromKey:kNXLB0];
172 }
173
174 + (NSString *)labelForOrderToolsNote:(NSUInteger)index
175 {
176     return [LABEL stringAtIndex:index fromKey:kNXLB0];
177 }
178
179 + (NSUInteger)countPattern:(NSString*)pattern inString:(NSString*)aString
180 {
181     NSRegularExpression *countRegex = [NSRegularExpression regularExpressionWithPattern:pattern options:0 error:&error];
182
183     return [countRegex numberOfMatchesInString:aString];
184 }
185
186 + (NSString*)recapViewFormatterAtIndex:(NSUInteger)index
187 {
188     NSString *formatter = [[self class] stringAtIndex:index];
189
190     NSInteger numberOfMatches = [[self class] countPattern:@"\\d" inString:[formatter stringByReplacingOccurrencesOfString:@"\\$" withString:@""]];
191     if (numberOfMatches == 2) {
192         return [self renderSubLabelsForLabel:formatter];
193     }
194     *error = [NSError errorWithDomain:kNXLBErrorWrongFormat code:1];
195     return nil;
196 }
197
198
199 + (NSString *)collectionHistoryUnreadFormatter
200 {
201     NSString *formatter = kNXLBCollectionHistoryUnreadFormatter;
202
203     NSUInteger numberOfMatches = [[self class] countPattern:@"\\d" inString:[formatter stringByReplacingOccurrencesOfString:@"\\$" withString:@""]];
204     if (numberOfMatches == 2) {
205         return [self renderSubLabelsForLabel:formatter];
206     }
207     return formatter;
208 }
209
210
211 + (NSArray *)controllersWithValue:(NSString *)value
212 {
213     NSString *sectionValue = [LABEL stringForKey:value];
214     if (!sectionValue) {
215         PLog(@"Section not found for key: %@", value);
216         return nil;
217     }
218
219     NSPredicate *tabBarPredicate = [NSPredicate predicateWithFormat:@"SELF.%@ = %@", sectionValue, value];
220     return [GET_APP_DELEGATE().tabBarController.viewControllers filter:tabBarPredicate];
221 }
222
223
224 + (UIViewController *)controllerWithValue:(NSString *)value
225 {
226 }
```

Instruments

Analizzatore di performance:

- Memory allocations
- Leaks
- Network activity
- CPU usage
- ...

Tool disponibile con Xcode



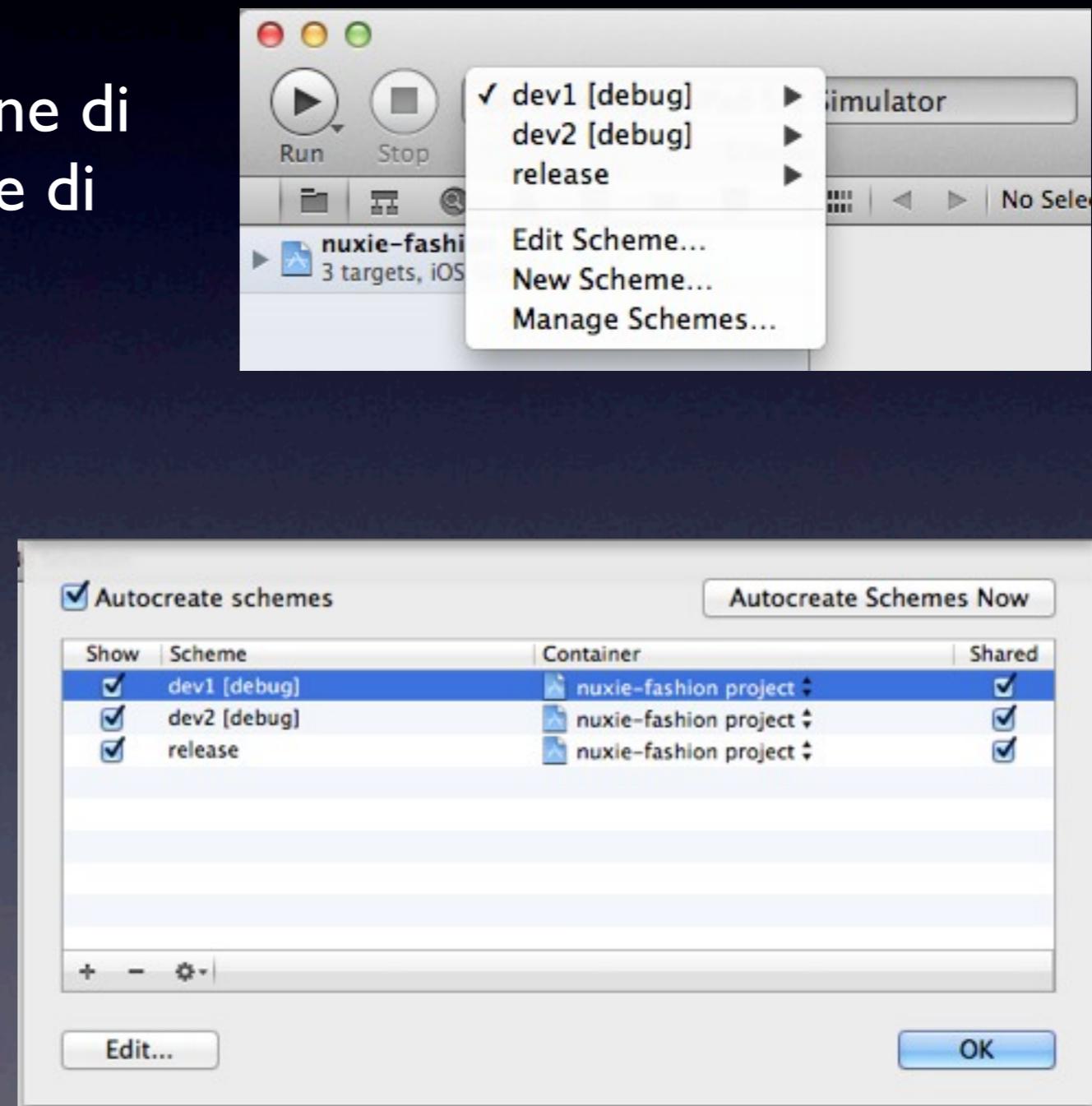
Scheme, Target & Workspace

Development flow

- Xcode Targets & Schemes
- Projects and Workspaces
- Organizer
- Build on device
- iOS Developer Program
- Provisioning Portal
- Certificates & Provisioning Profiles
- iTunes Connect & deploy on the App Store

Schemes

Gli schemi di Xcode definiscono una collezione di target, di configurazioni e di test da usare durante la compilazione.



Targets 1/2

Un target specifica un'applicazione (prodotto) da compilare e contiene le istruzioni di compilazione a partire da un insieme di file in un progetto o in un workspace.

Un target definisce un singolo prodotto. Un progetto può contenere più target ognuno dei quali genera un prodotto (potenzialmente diverso).

Le informazioni necessarie per compilare un prodotto prendono il nome di ‘build settings’ e ‘build phases’ ispezionabili e modificabili dal progetto Xcode.

Targets 2/2

Solo un target è attivo allo stesso momento (specificato dallo schema).

Il prodotto compilato da un target può essere relazionato a un altro target: in questo caso si dice che il secondo target ha una dipendenza verso il primo. Tale dipendenza può essere rilevata da Xcode se entrambi i target sono presenti all'interno dello stesso workspace.

Project and Workspace

Un progetto è un insieme di file, risorse e informazioni necessarie per compilare uno o più prodotti software.

Il workspace, introdotto con Xcode 4, è uno spazio di lavoro comprendente uno o più progetti. E' utile per condividere risorse tra progetti e migliora l'organizzazione di tutto ciò che include.

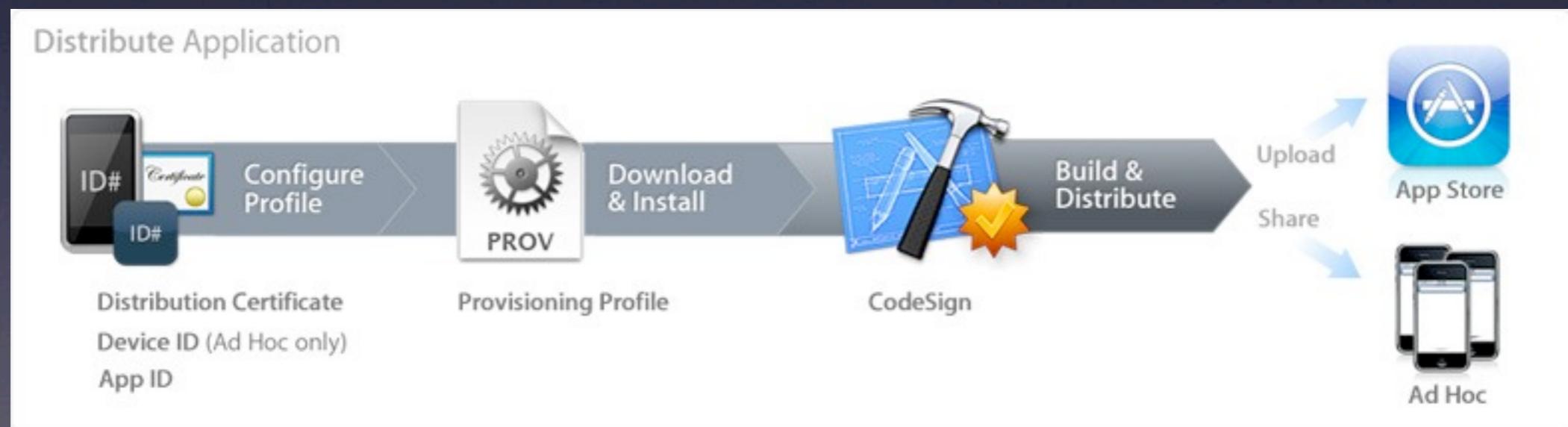
Publish flow & Co.

iOS Developer Program

- Per distribuire applicazioni (su App Store o Ad Hoc) è necessaria una licenza a pagamento (iOS Developer Program).
- La licenza per sviluppatori (\$99/year) permette di:
 - Compilare ed eseguire su device
 - Abilitare fino a 100 device per lo sviluppo o beta testing
 - Distribuire applicazioni Ad Hoc (ai device abilitati)
 - Distribuire applicazioni su App Store

Provisioning Portal

- Nel Developer Portal è possibile creare e gestire i certificati necessari per firmare le applicazioni.
- Ogni app deve essere firmata per essere eseguita su device
- Sia in sviluppo sia in ambienti di produzione
- Guide esaurenti nel Provisioning Portal



iTunes Connect

Nell'iTunesConnect è possibile gestire ogni aspetto della presenza su App Store:

- pubblicare e gestire applicazioni su App Store
- gestire i contratti legali per la vendita di applicazioni
- controllare il numero di download e le vendite
- consultare i report mensili sui guadagni
- gestire iAD
- gestire gli utenti abilitati a visionare i dati di cui sopra
- visualizzare i crash logs
- ...

**Welcome, Alberto De Bortoli**

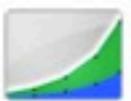
iTunes Connect provides tools to help manage your content in the App Store.

NEW

We've updated the [App Store Marketing and Advertising Guidelines for Developers](#) with instructions on using the App Store badge—including new localized versions in Simplified and Traditional Chinese, Japanese, and Portuguese—as well as featuring Apple products in marketing materials, creating custom photography and video, promoting apps with accessories, and more.

NEW

The App Store is now on [Facebook](#) and [Twitter \(@AppStore\)](#). Follow along for App Store recommendations, exclusive offers, tips, tutorials, and more. You can like, comment, share, and retweet content, spreading the word to your own marketing outlets, fans, and followers.

**Sales and Trends**

Preview or download your daily and weekly sales information here.

**Manage Your Applications**

Add, view, and manage your applications in the iTunes Store.

**Contracts, Tax, and Banking**

Manage your contracts, tax, and banking information.

**iAd Network**

View ad performance and manage the ads that appear in your apps.

**Payments and Financial Reports**

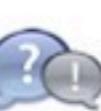
View and download your monthly financial reports and payments.

**Catalog Reports**

Request catalog reports for your App Store content.

**Manage Users**

Create and manage both iTunes Connect and In App Purchase Test User accounts.

**Developer Forums**

Find solutions and share tips with Apple developers from around the world.

**Contact Us**

Having a problem uploading your application? Can't find a Finance Report? Use our Contact Us system to find an answer to your question or to generate a question to an iTunes Rep.

[Download the Developer Guide.](#)[FAQs](#) Review our answers to common inquiries.

iOS Enterprise Program

- La licenza enterprise (\$299/year) permette di:
 - Compilare ed eseguire su device
 - Installare l'applicazione su qualsiasi device, ma chi lo fa deve essere “legalmente” collegato all'azienda proprietaria del certificato (es: dipendenti, agenti di vendita, ecc...)
 - Distribuire applicazioni con installazione Over The Air
- Caso d'uso: ogni cliente di H-umus ha il proprio certificato enterprise

Riferimenti

- Iniziamo ridendo :-) <http://www.textfromxcode.com/>
- **iOS Recipes: Tips and Tricks for Awesome iPhone and iPad Apps** (M. Drance, P. Warren, The Pragmatic Programmers)
- **iOS 5 Programming Cookbook** (V. Nahavandipoor, O'Reilly)
- **Objective-C: Developer Reference** (J. DeVoe, Wiley)
- **Pro Objective-C Design Patterns for iOS** (C. Chung, Apress)
- **Matt Gallagher:**
<http://cocoawithlove.com/>
- **Cocoa is my girlfriend:**
<http://www.cimgf.com/>
- **Ray Wenderlich site:**
<http://www.raywenderlich.com>
- **mikeash.com: just this guy, you know?:**
<http://www.mikeash.com/pyblog>
- Last but not least...
developer.apple.com Documentazione Apple

