CocoaHeads

Thurs May 15, 2008

These Things First

1 iPhones to the silent position

2 NSCoder Night

3 WWDC Meeting: Wed, June 11

4 Joel Norvell: PDF Forms in Cocoa

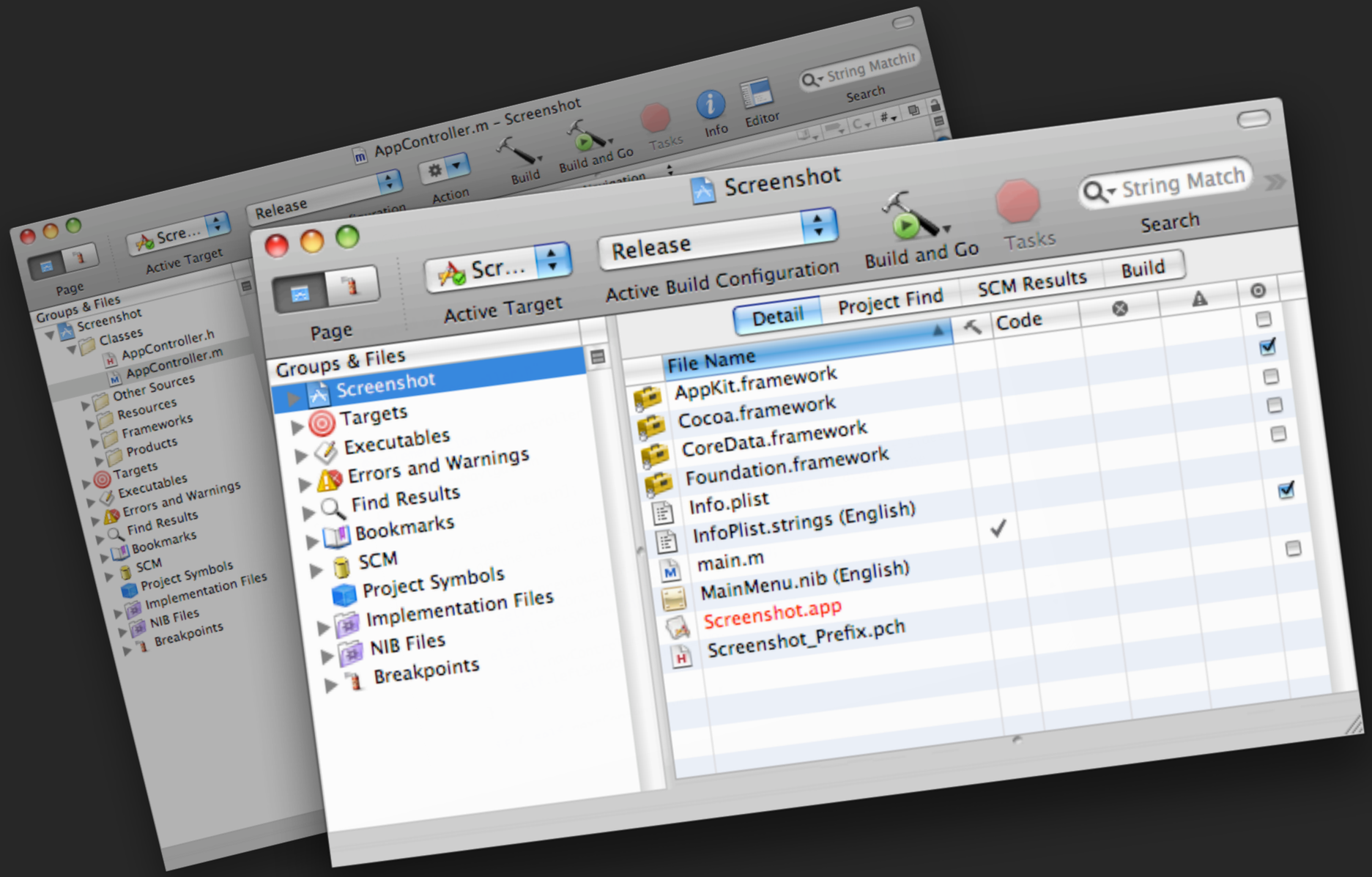
5 Scott Stevenson: Best of Both Worlds

Best of Both Worlds

Scott Stevenson

Roadmap

Xcode	Objective-C	Design
Interface Builder	Mac OS X	Culture
		Advanced


AppController.m – Screenshot
Page
Active Target
Release
Action
Build
Build and Go
Tasks
Info
Editor
String Matchir
Search
Groups & Files
Screenshot
Classes
AppController.h
AppController.m
Other Sources
Resources
Frameworks
Products
Targets
Executables
Errors and Warnings
Find Results
Bookmarks
SCM
Project Symbols
Implementation Files
NIB Files
Breakpoints
Screenshot
Active Build Configuration
String Match
Detail
Project Find
SCM Results
Build
Code
File Name
AppKit.framework
Cocoa.framework
CoreData.framework
Foundation.framework
Info.plist
InfoPlist.strings (English)
main.m
MainMenu.nib (English)
Screenshot.app
Screenshot_Prefix.pch

Xcode

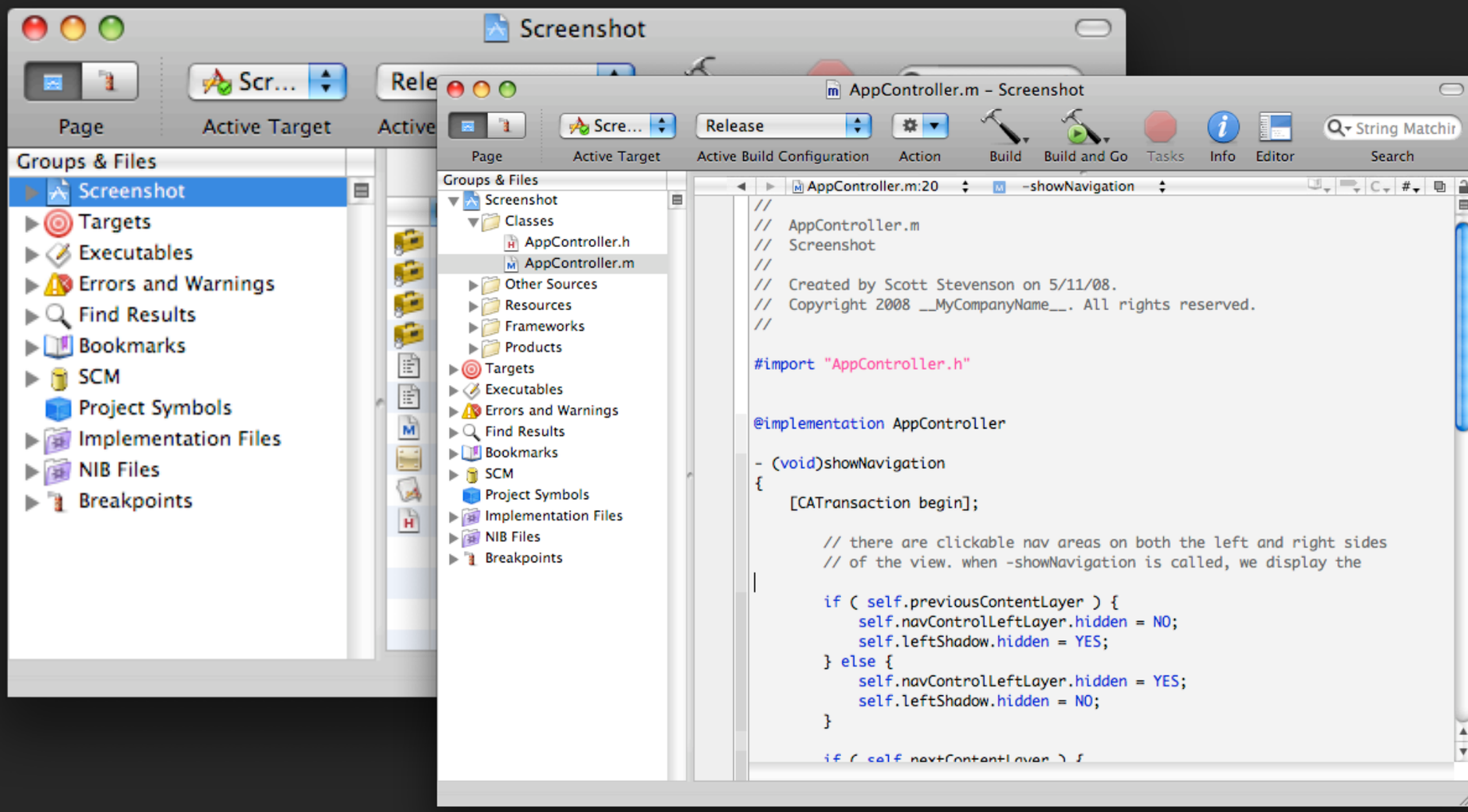
Xcode

Projects, editing, source control

Uses gcc and gdb

Written with Cocoa, used by Apple

Third-party integration



Features

Templates

Code Completion

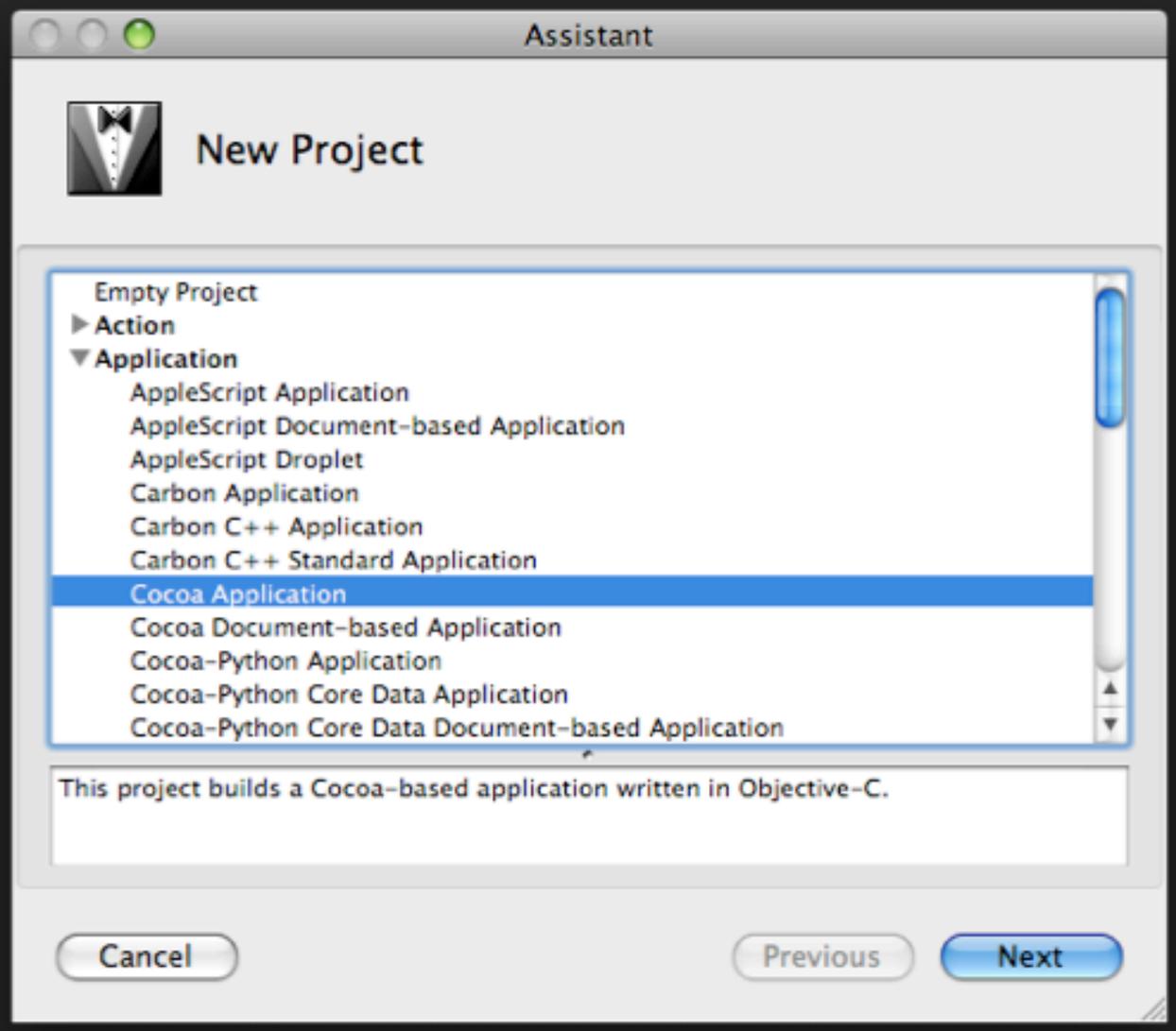
Inline Feedback

Debugging

Documentation

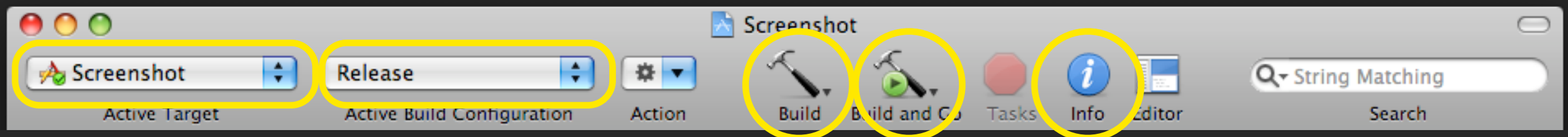
Visual gcc interface

Refactoring

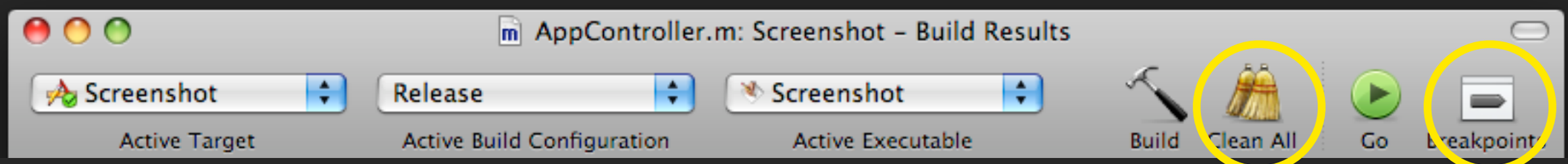


Toolbars

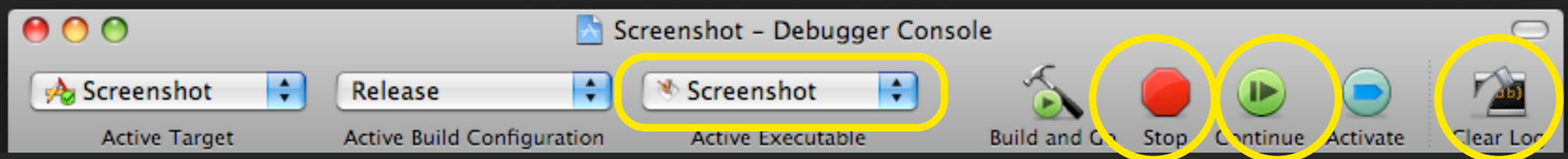
Project View



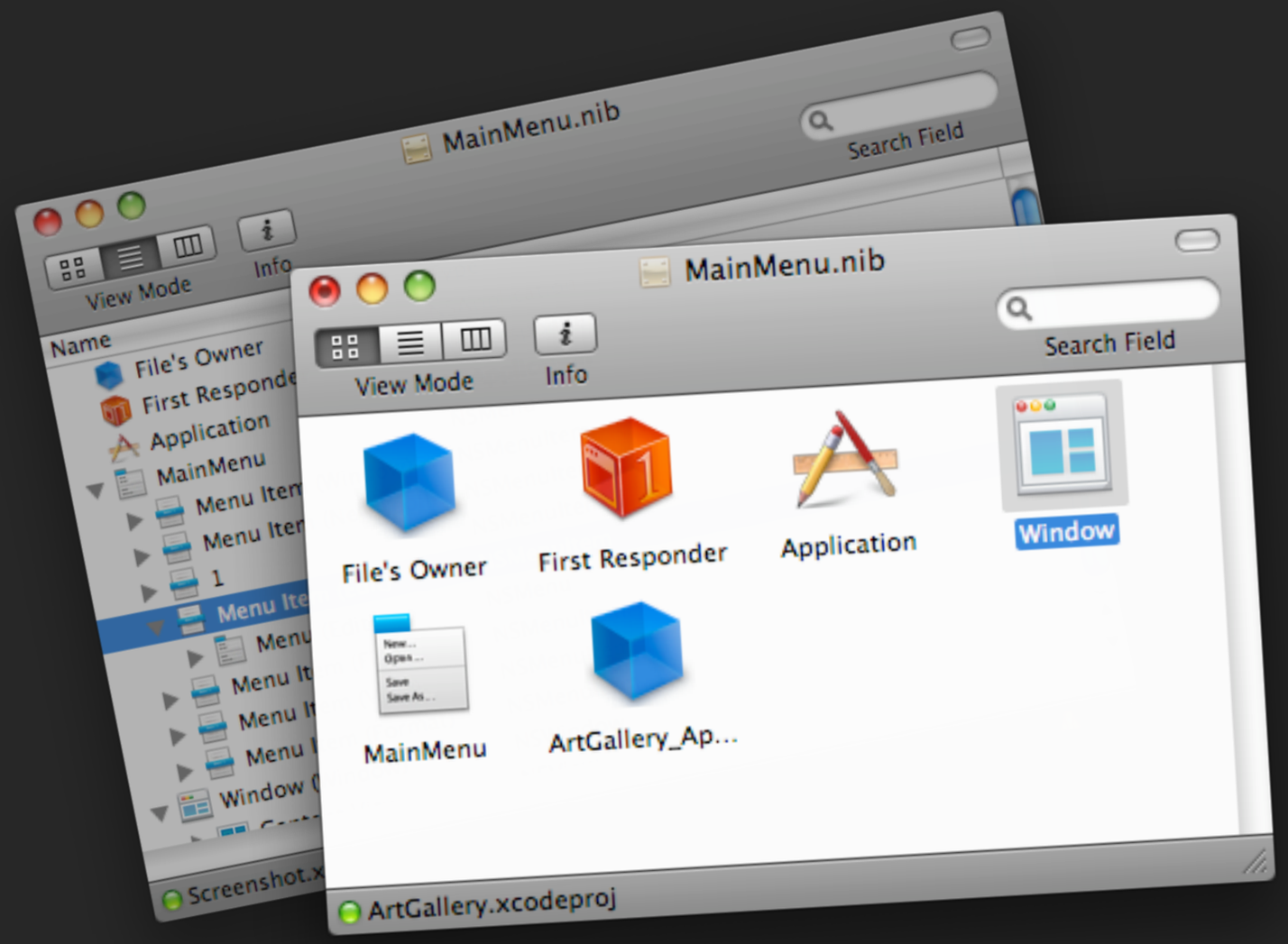
Build Results



Debugger



Xcode Tour



Interface Builder

Interface Builder

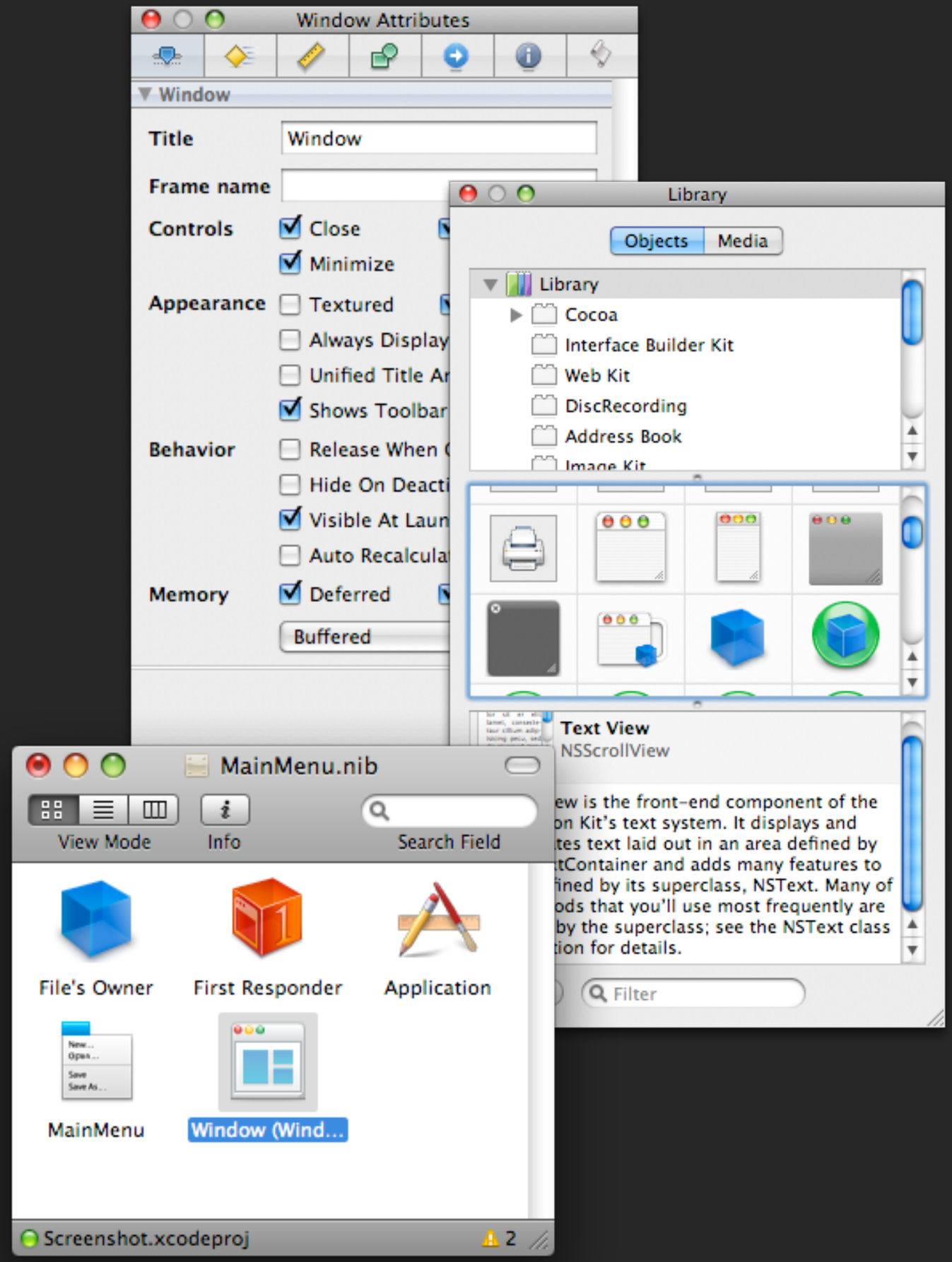
Basic layout and setup

Prototyping

Visualizing

XIB and NIB files

Integrated with Xcode



Interface Builder Tour

AppController.m

AppController.m:17 -showNavigation

```objc
#import "AppController.h"

@implementation AppController

- (void)showNavigation
{
    [CATransaction begin];

        if ( self.previousContentLayer ) {
            self.navControlLeftLayer.hidden = NO;
            self.leftShadow.hidden = YES;
        } else {
            self.navControlLeftLayer.hidden = YES;
            self.leftShadow.hidden = NO;
        }

        if ( self.nextContentLayer ) {
            self.navControlRightLayer.hidden = NO;
            self.rightShadow.hidden = YES;
        } else {
            self.navControlRightLayer.hidden = YES;
            self.rightShadow.hidden = NO;
        }

    [CATransaction commit];
}
@end
```

Objective-C

Objective-C

Primary language for Cocoa

Object additions to C

Dynamic runtime

Weakly typed

Simple syntax

Best-kept secret

Can integrate with C++

Memory Management

Garage Collection on 10.5

Reference counting

Do not free object memory

Core Foundation

AppController.m

AppController.m:17 -showNavigation

```
#import "AppController.h"


@implementation AppController

- (void)showNavigation
{
    [CATransaction begin];

        if ( self.previousContentLayer ) {
            self.navControlLeftLayer.hidden = NO;
            self.leftShadow.hidden = YES;
        } else {
            self.navControlLeftLayer.hidden = YES;
            self.leftShadow.hidden = NO;
        }

        if ( self.nextContentLayer ) {
            self.navControlRightLayer.hidden = NO;
            self.rightShadow.hidden = YES;
        } else {
            self.navControlRightLayer.hidden = YES;
            self.rightShadow.hidden = NO;
        }

    [CATransaction commit];
}
@end
```

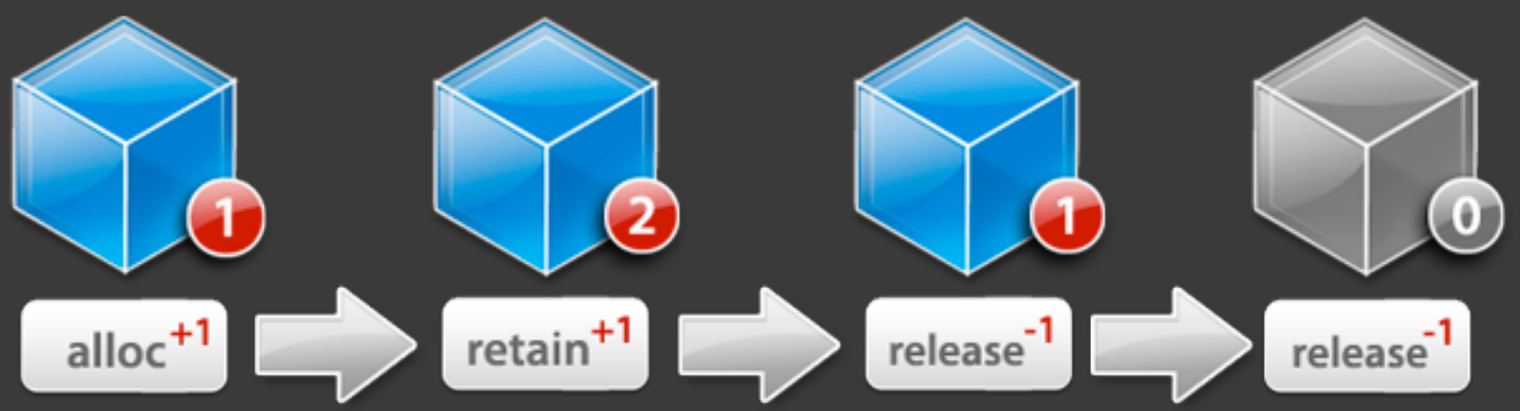
Memory Management

Object starts with 1

Increase with -retain, -copy

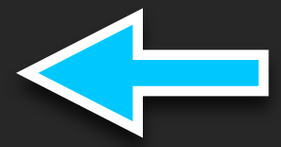
Decrease with -release, -autorelease

Final release triggers -dealloc



Runtime

Create methods at runtime



Intercept/redirect messages

Missing methods are warnings

Runtime loading of plug-ins

You may be subclassed at runtime

Typing

Objective-C is weakly typed: id

Toll-free bridging

- NSString : CFStringRef
- NSArray : CFArrayRef
- NSDictionary : CFDictionaryRef

Messages

Not direct method calls

Can be perform delayed

```
NSString* value = [textField stringValue];

[textField setStringValue:@"Your Name"];
[textField setValue:name forKey:kNameKey];

[textField setValue:name
             forKey:kNameKey];

[nil setValue:name forKey:kNameKey];
```

Messages

Alternate syntax for accessors

```
NSString* value = textField.stringValue;
textField.stringValue = @"Your Name";
```

Only for setters and getters

Not direct ivar access

Classes

Separate header and implementation files

Single inheritance

```
@interface MyClass: NSObject {
    NSString* title;
    NSDate* creationDate;
}
- (NSString*) title;
- (NSDate*) creationDate;
- (void) setCreationDate: (NSDate*)newDate;
- (void) resetCreationDate;
- (void) setTitle: (NSString*)newTitle;
@end
```

```
@implementation MyClass

- (id) init
{
   if ( self = [super init] )
   {
      title = nil;
   }
   return self;
}
```

```objc
@implementation MyClass

- (NSString*) title {
   return title;
}

- (void) setTitle: (NSString*)newTitle {
   [title autorelease];
   title = [newTitle retain];
}

- (void) dealloc {
   [title release];
   [super dealloc];
}
@end
```

Classes

Creating objects

```
MyClass* object = [[MyClass alloc] init];

[object setTitle:title];
[object setCreationDate:[NSDate date]];
[object release];
```

Properties

```
@interface MyClass: NSObject {
   NSString* title;
}
@property (retain) NSString* title;
@end
```

```
@implementation MyClass
@synthesize title;

- (id) init {
   self.title = nil;
}
@end
```

Mac OS X

Mac OS X

Application Packages

Installation

Respect the user's space:

- ~/Library/Application Support
- ~/Library/Preferences

Do not put anything in:

- Home
- Documents
- /System/
- Hidden Directories

App Package

Do not change contents

Essentials: keep self-contained

Application Support

Real data for non-document apps

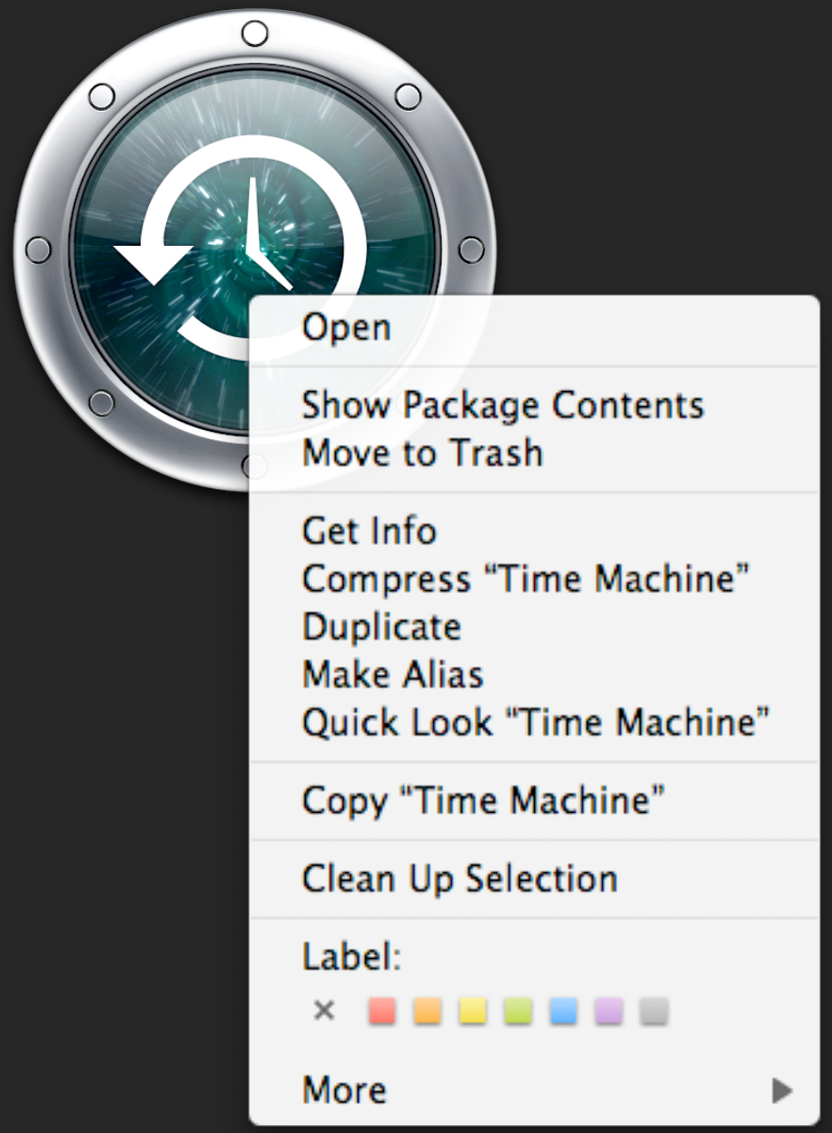
Third Party Plug-ins

Preferences

Disposable

Standard format

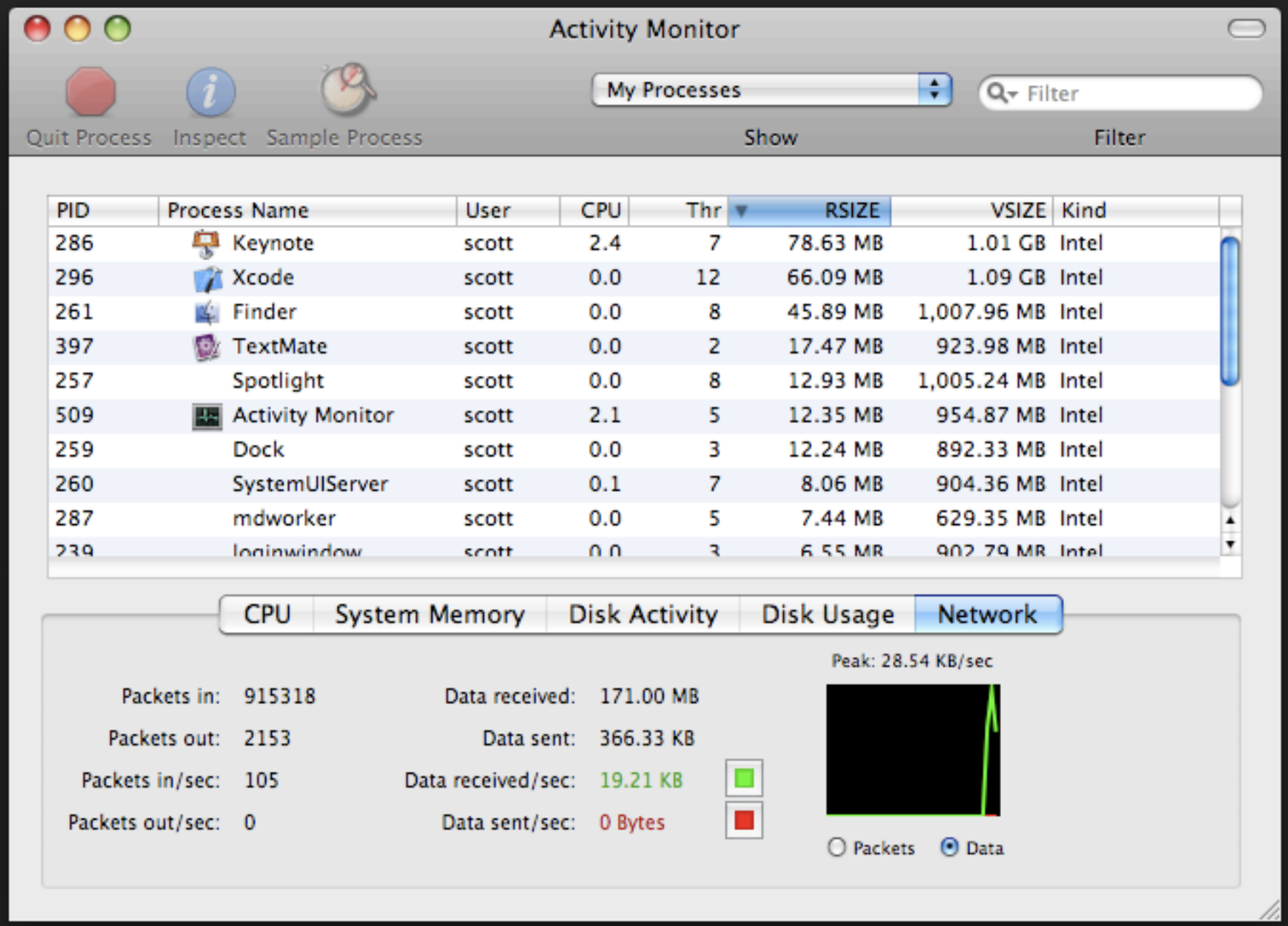
Delete during testing



Background Processes

Don’t do it.

Seriously, don’t do it.



Design

Developer

The appearance and behavior of the view layer.

User

The product.

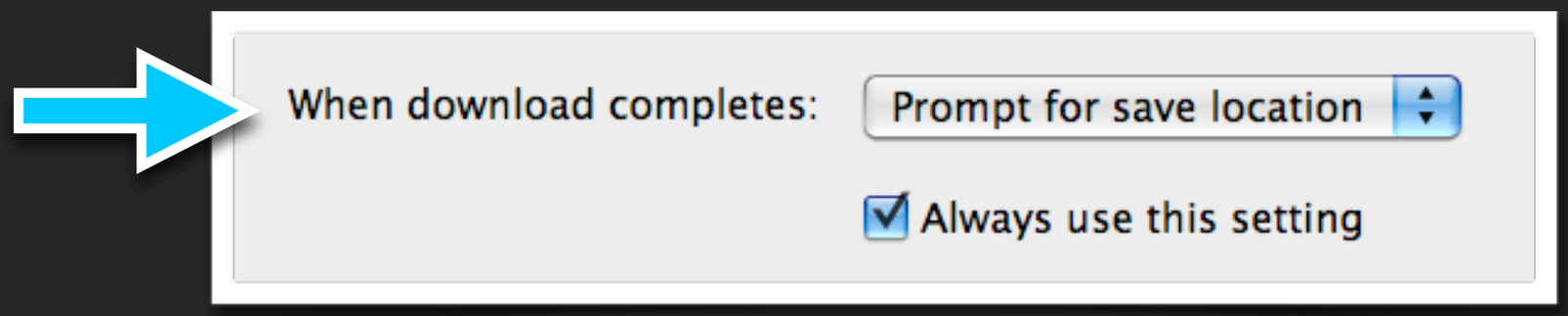

AT&T
9:42
Sunday, June 3
slide to unlock

Labels and Prompts

Use common words and complete sentences.

Don’t use lingo or clever language.

Prompt only for multiple items.
Labels for everything else.



Simplicity and clarity win every time.

Model prefs design citizen: Safari

Icons

First impressions

Specialized Skill

Full size

Appealing

Functional and conceptual

Common Mistakes

Splash screens

Inventing your own UI style

Throwing files everywhere

Too much user interface

Bizarre font choices

If You Do Nothing Else

Keep it simple.

Culture

Code Culture

Use clear method names

Avoid subclassing

Learn the frameworks

Trust the frameworks

Use the highest level abstractions

User Culture

Mac developers live online

Users buy into *you*

Respect your users

Mac news sites

Apple Top Downloads

WWDC

CocoaHeads (your are here!)

Wrap Up

Wrap-up

http://theocacao.com/