

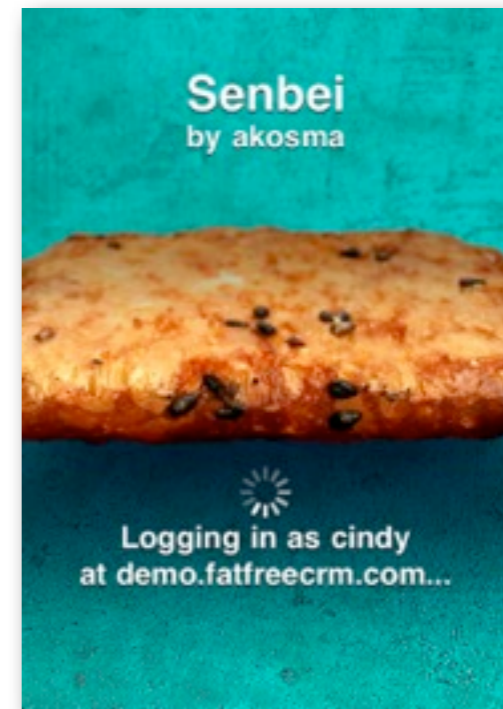
Ten Commandments

for iPhone Software Development

Adrian Kosmaczewski

akosma software

akosma.com
github.com/akosma
[linkedin.com/in/akosma](https://www.linkedin.com/in/akosma)
formspring.me/akosma
twitter.com/akosma
slideshare.com/akosma



Some questions

Who's new to iOS?

Which technologies?

J2EE

J2ME

.NET

Ruby / Rails

others?

**Which programming
languages?**

C / C++?

Java, C#?

Ruby, Python, Lua?

JavaScript?

Fortran, Lisp, COBOL?

10 Commandments



<http://www.flickr.com/photos/oseillo/345879263/>



<http://www.flickr.com/photos/justdrew1985/4348527596/>

Jérôme Commandeur



<http://akos.ma/gib5>

1

**Thou shalt manage
memory properly**



- iPhone 3G: 128 MB RAM
- iPhone 3GS, iPad: 256 MB RAM
- iPhone 4: 512 MB RAM

±70 MB for the OS!

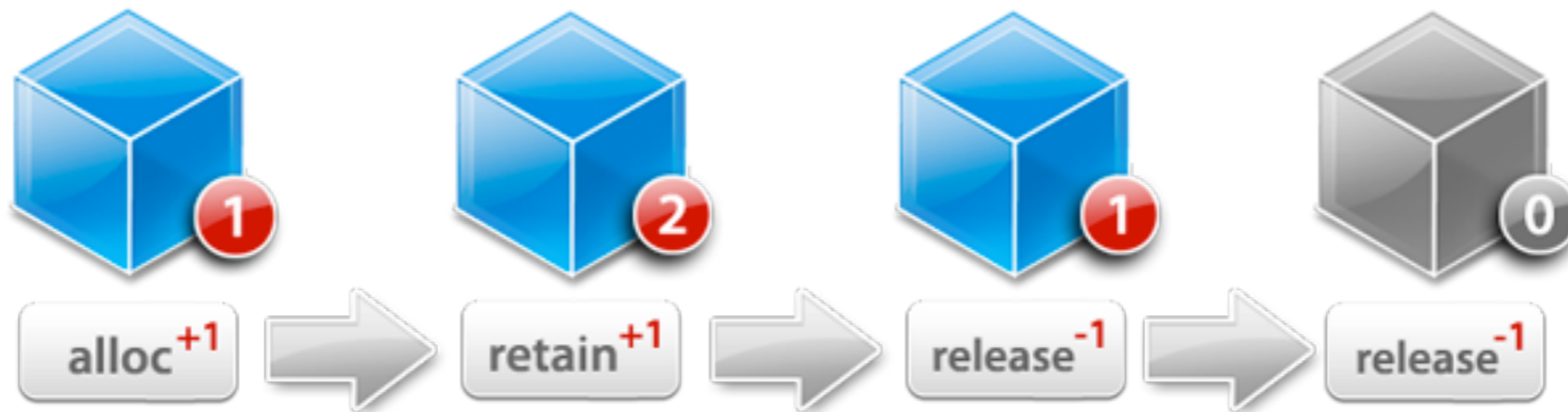
no swap file

(no virtual memory)



no garbage collection

objects have a
“retain count”



basic rule:

for every
[alloc], [retain], [copy]

there must be a
[release]

beware:

**Objective-C only allows
objects on the heap**



<http://linguiniontheceiling.blogspot.com/2008/10/thats-madame-trash-heap-to-you.html>

**No automatic objects
on the stack (C++)**



```
// C++  
// Memory freed when out of scope  
std::string name("Adrian");  
  
std::string *name = NULL;  
name = new std::string("Adrian");  
delete name;
```

iPhone OS memory warnings

```
- (void) didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
}

- (void) applicationDidReceiveMemoryWarning:(UIApplication *)application
{
    [[ImageCache sharedImageCache] removeAllImagesInMemory];
}

NSNotificationCenter *center = [NSNotificationCenter defaultCenter];
[center addObserver:self
             selector:@selector(whatever:)
             name:UIApplicationDidReceiveMemoryWarningNotification
             object:nil];
```




TOP TAGS

App Store 14	conference 13	Objective-C 12	Switzerland 13
Apple 12	iPad 12	open source 10	Video 12
Business 12	iPhone 61	software 12	All Tags 175

JAN 28, '09
2:00 PM

10 iPhone Memory Management Tips

AUTHOR **Adrian Kosmaczewski**

CATEGORIES **Tech, iPhone**

TAGS **Cocoa, How to?, iPhone, Objective-C**

[EDIT](#)

Memory management in the iPhone is a hot topic. And since tonight I'm talking about it on tonight's monthly meetup of the **French-speaking Swiss iPhone Developers group**, I might as well share some tips here from my own experience.

I won't go dive through the basics; I think that Scott Stevenson did a great job in his **"Learn Objective-C" tutorial at CocoaDevCentral**, from where the image below comes. I'm just going to highlight some iPhone-specific issues here and there, and provide some hints on how to solve them.



To begin with, some important background information:

- The iPhone 3G has 128 MB of RAM, but at least half of it might be used by

CATEGORIES

CONTACT

From 0800 to 1800 CET
Tel: + 41 78 739 47 76
Mail: software@akosma.com
AIM: [akosma@mac.com](aim:akosma@mac.com)
Skype: [akosma](#)
Twitter: [@akosma](#)

TWITTER

Going to Geneva. Time for Dev Day for iPhone
<http://akos.ma/t4y>
wooooo!
12 mins ago
[More updates...](#)

SHARE

[del.icio.us](#)
[Digg](#)
[Evernote](#)
[Facebook](#)
[FriendFeed](#)
[Google](#)

PARTNERS

Coriolis Technologies
Creates custom software for the iPhone and for the Apple Macintosh, in C/C++ or Cocoa.

Devego
They can make you an awesome website. They are devoted to create websites that are stunning and easy to use.

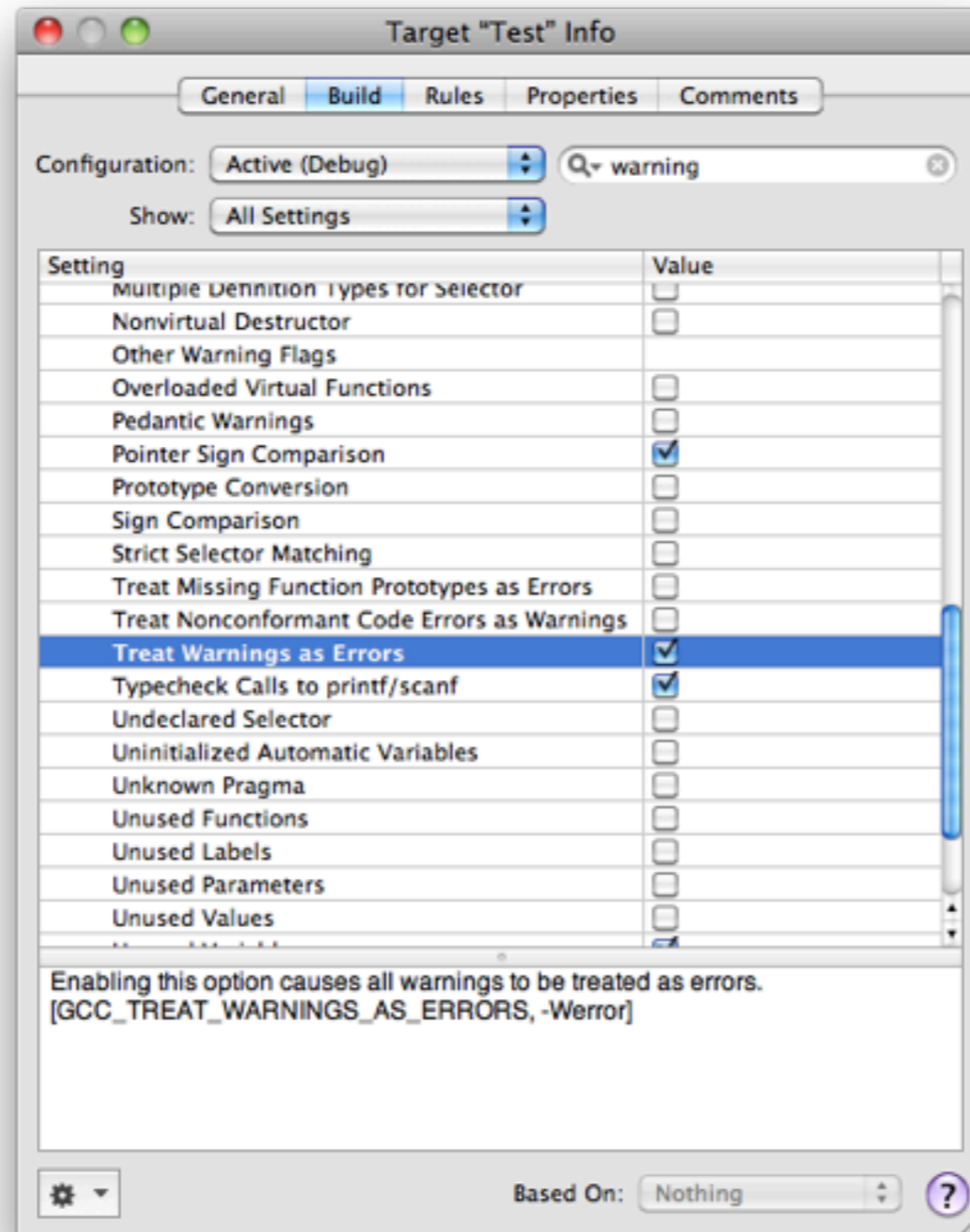
Infobahn
Infobahn creates and deploys internet, extranet and intranet sites using their own content management system EZMODIFIER™

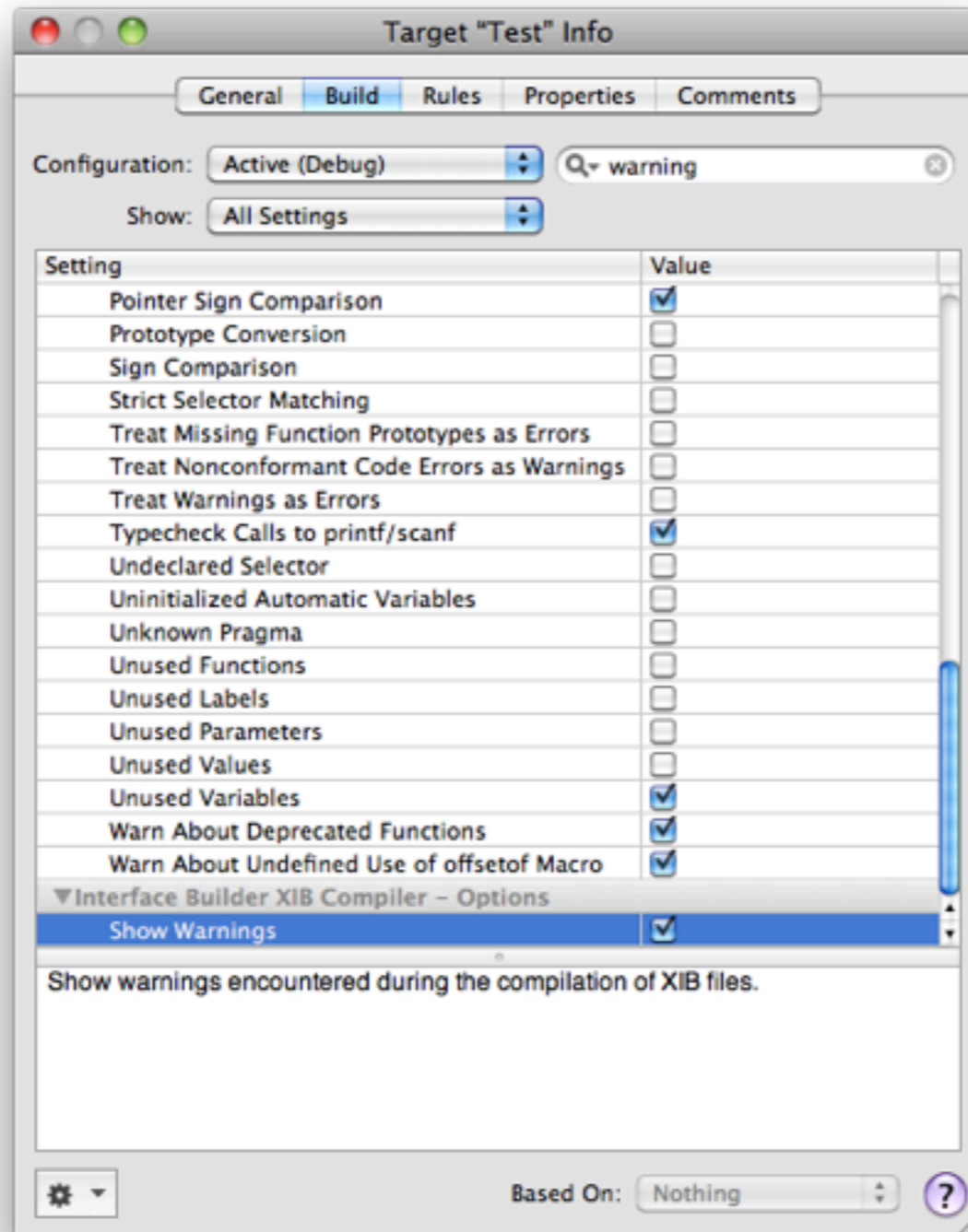
Trifork A/S
Trifork develops, sells and delivers IT solutions to the financial and public sector in Denmark. Trifork A/S has 130 employees distributed in offices in Aarhus, Copenhagen, California and Switzerland.

Zerofee
Ethical design agency – they create visual identity and design for print and digital media, but not for irresponsible brands or companies.

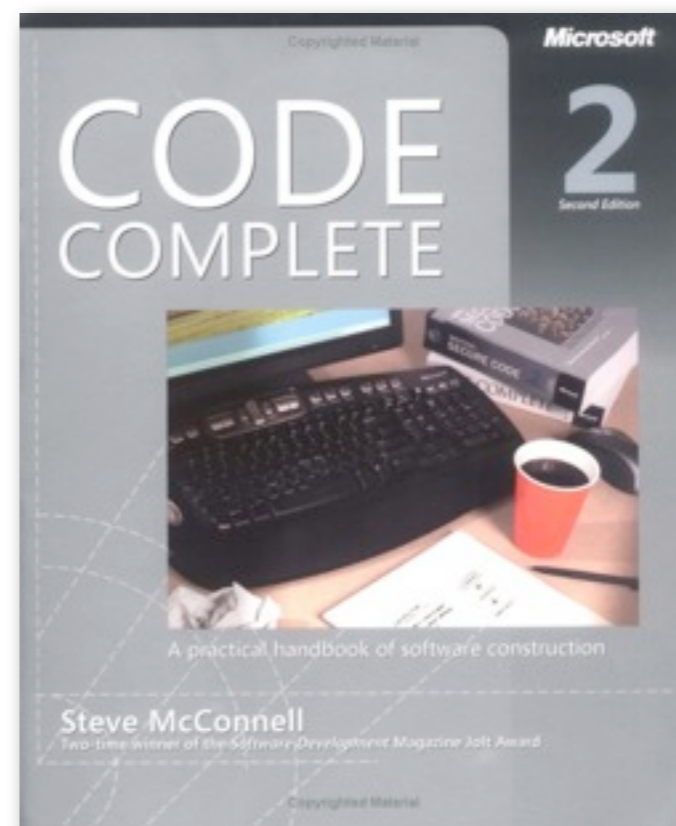
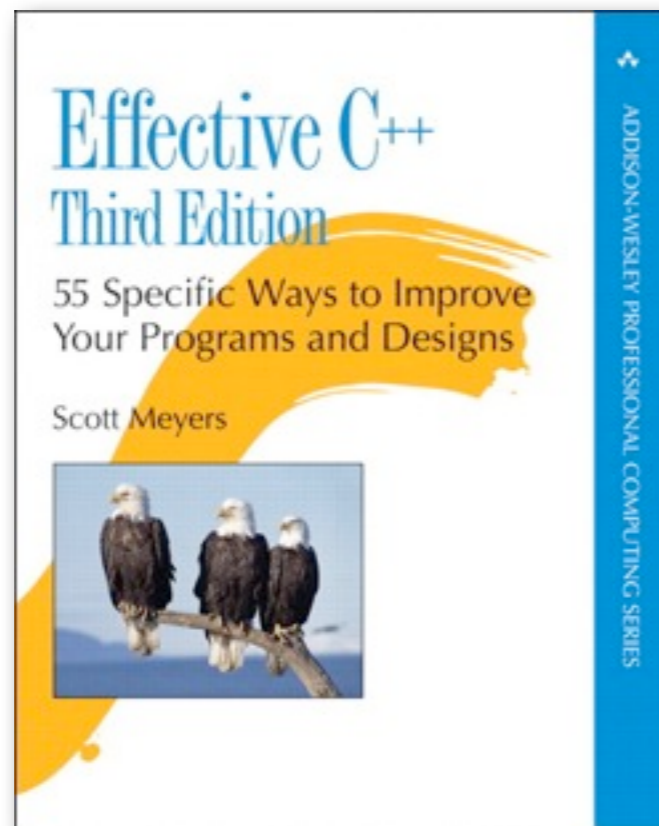
2

**Thou shalt remove all
compiler warnings**





GCC_TREAT_WARNINGS_AS_ERRORS
-Werror



Why Warnings?

- Using deprecated symbols;
- Calling method names not declared in included headers;
- Calling methods belonging to implicit protocols;
- Forgetting to return a result in methods not returning “void”;
- Forgetting to `#import` the header file of a class declared as a forward “`@class`”;
- Downcasting values and pointers implicitly.

Solutions

**Make your intentions
explicit to the compiler**

- Make implicit protocols explicit
- Create categories for private methods
- Turn implicit type conversions and casts into explicit ones
- Use `@class` in the `@interface`, `#import` on the `@implementation`

TOP TAGS

App Store	14	conference	13	Objective-C	12	Switzerland	13
Apple	12	iPad	12	open source	10	Video	12
Business	12	iPhone	61	software	12	All Tags	175

JUL 16, '09
2:11 PM

Objective-C Compiler Warnings

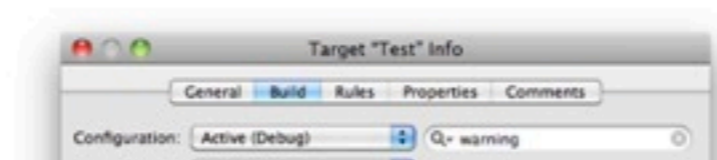
AUTHOR **Adrian Kosmaczewski**
CATEGORIES **Tech, iPhone**
TAGS **Cocoa, Opinion, Tech**

[EDIT](#)

A recent comment by [Joe D'Andrea](#) in a [previous post](#) reminded me about the importance of removing compiler warnings in Xcode projects. Most importantly, it reminded me of a conversation with a fellow developer a couple of weeks ago, in which he told me that he was surprised to see that my projects compiled all the time without warnings. Not a single one. Nada. And that I took the time to remove them before checking code into source control.

He actually didn't know you could remove all compiler warnings; he thought Objective-C was the land of compiler warnings. This situation, I think, is far from exceptional, and due mostly to cultural and technical reasons.

It is my opinion, that removing compiler warnings is **basic project hygiene**, like writing unit tests, or using the Clang Static Analyzer. I will explain in this post some techniques I use to remove warnings in my Objective-C code.



CATEGORIES

CONTACT

From 0800 to 1800 CET
Tel: **+ 41 78 739 47 76**
Mail: software@akosma.com
AIM: [akosma@mac.com](aim:akosma@mac.com)
Skype: **akosma**
Twitter: **@akosma**

TWITTER

Going to Geneva. Time for Dev Day for iPhone

<http://akos.ma/t4y>

wooooo!

12 mins ago

[More updates...](#)

SHARE

- [del.icio.us](#)
- [Digg](#)
- [Evernote](#)
- [Facebook](#)
- [FriendFeed](#)
- [Google](#)

PARTNERS

Coriolis Technologies

Creates custom software for the iPhone and for the Apple Macintosh, in C/C++ or Cocoa.

Devego

They can make you an awesome website. They are devoted to create websites that are stunning and easy to use.

Infobahn

Infobahn creates and deploys internet, extranet and intranet sites using their own content management system EZMODIFIER™

Trifork A/S

Trifork develops, sells and delivers IT solutions to the financial and public sector in Denmark. Trifork A/S has 130 employees distributed in offices in Aarhus, Copenhagen, California and Switzerland.

Zerofee

Ethical design agency – they create visual identity and design for print and digital media, but not for irresponsible brands or companies.

3

Honor the Human Interface Guidelines

[http://developer.apple.com/iphone/
library/documentation/
userexperience/conceptual/
mobilehig/](http://developer.apple.com/iphone/library/documentation/userexperience/conceptual/mobilehig/)

Introduction

Part I: Planning Your iPhone Software Product

- ▶ The iPhone OS Platform: Rich with Possibilities
 - ▶ Human Interface Principles: Creating a Great User Interface
 - ▶ Designing an iPhone Application: From Product Definition to Branding
 - ▶ Handling Common Tasks
- Part II: Designing the User Interface of Your iPhone Application**
- ▶ A Brief Tour of the Application User Interface
 - ▶ Navigation Bars, Tab Bars, Toolbars, and the Status Bar
 - ▶ Alerts, Action Sheets, and Modal Views
 - ▶ Table Views, Text Views, and Web Views

- ▶ Application Controls
- ▶ System-Provided Buttons and Icons
- ▶ Creating Custom Icons and Images

Revision History

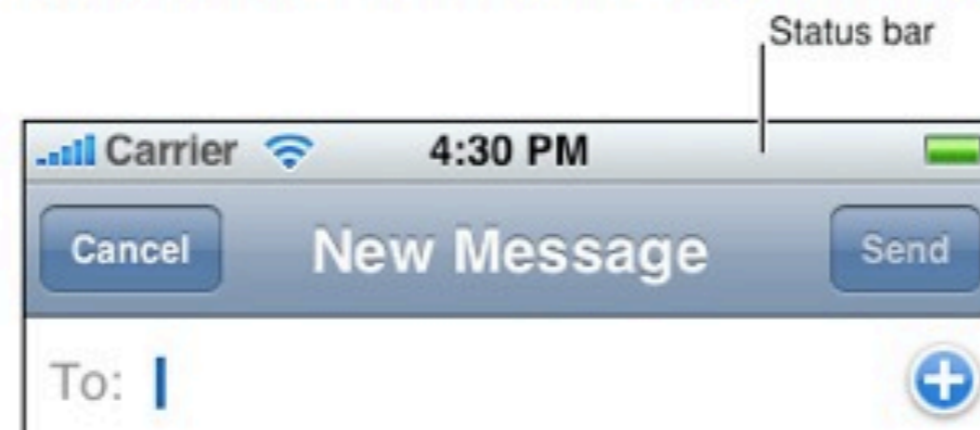
Navigation Bars, Tab Bars, Toolbars, and the Status Bar

The status bar, navigation bar, tab bar, and toolbar are views that have specifically defined appearances and behaviors in an iPhone application. These bars are not required to be present in every application (immersive applications often don't display any of them), but if they are present, it's important to use them correctly. The reason is that these bars provide familiar anchors to users of iPhone OS-based devices, who are accustomed to the information they display and the types of functions they perform.

The Status Bar

The status bar shows users important information about their device, including cell signal strength, the current network connection, and battery charge. Figure 6-1 shows an example of a status bar.

Figure 6-1 A status bar contains important information for users



Developing and Designing Cocoa Touch Applications



Programming the

iPhone User Experience

O'REILLY®

Toby Boudreaux

Your Objective:

avoid rejections



News and Announcements for iPhone Developers

As the iPhone Developer Program has grown and new developers come to the platform, so has the need to communicate information, guidance, and tips more frequently. We are pleased to introduce News and Announcements for iPhone Developers which will provide you with these updates on a regular basis.

Sep 28, 2009

A Special Thanks to the iPhone Developer Community

Today, we [announced](#) that more than two billion apps have been downloaded from the App Store.

We'd like to take this opportunity to thank you, the iPhone developer community, as we mark this milestone. Your creativity and innovation have been instrumental in helping us to create a mobile experience that is unparalleled in the industry.



Sep 25, 2009

Setting Up an In App Purchase Test User in iTunes Connect

If your application takes advantage of In App Purchase, you must set up an in app purchase test user in iTunes Connect. To do so, log in, access the Manage Users module, select the In App Purchase Test User section, then follow these steps:



App Store Review Status

Based on the current volume of app submissions, 96% of applications are being approved within 14 days.



Last Updated: Sep 25, 2009

App Store Submission Tips

This series of tips provides you with guidance on the app submission and approval process [Read now](#) ▶

@sdegutis New rumors point to yes... it's getting close I think:

<http://idek.net/XEJ>

— 1 day ago [∞]

@danberte From what I've heard it will use a Mac OS X / iPhone OS hybrid...

— 1 day ago [∞]

If Apple doesn't re-imagine the iPhone OS homescreen before the tablet comes out, here's what will happen: <http://idek.net/XF6>

— 1 day ago [∞]

@Cocoaia They're gay and proud?

— 1 day ago [∞]

@boredzo Really? I figured it'd be terrible. Maybe I'll put it on Netflix



View my portfolio
Web design and other stuff



Subscribe to RSS
Tutorials, interviews, icons!

iPhone Application UI Design Patterns

[∞] Posted at 3:15am on a Tuesday in July — 13 Comments

Update: Changed the blog entry title to reduce confusion.

The iPhone is one big constraint — no keyboard, small screen, few buttons — so designing applications for the iPhone is an exercise in building smart, simple software. Bloated apps on the iPhone? You won't find many. Most applications pick one feature or group of related features and centralize the product around that central theme.

When Apple began crafting UIKit, the set of APIs used to build the user interface for an iPhone app, they had to see into the future and predict what the most common application design models would be and make sure those could be accomplished easily. It may seem obvious to us now because we're so used to iPhone application design but the high-level navigation and interaction concepts available to iPhone application developers are really quite brilliant:

- ▶ Dive deep into hierarchical levels of application information and then surface back to the top easily
- ▶ Switch between different main pieces of functionality without losing your place on one when moving to another
- ▶ Edit and adjust information without losing your place contextually
- ▶ Display a list of information or choices

HOME

GRAPHICS

SHOWCASES

INSPIRATION

FONTS

HOW-TO

CSS

TOO



Design book written
specifically for you!

Save 20%!

PRE-ORDER NOW!

iPhone Apps Design Mistakes: Over-Blown Visuals

🕒 By Alexander Komarov, July 21st, 2009 in [How-To](#) | [117 Comments](#) | [Forum](#)

<http://www.smashingmagazine.com/2009/07/21/iphone-apps-design-mistakes-overblown-visuals/>



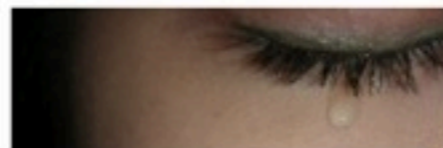
The iPhone App Developers' Blog: iPhone Programming, Developer News, Interviews And T

Jobs: [Storm8 is looking for a fulltime iPhone and/or Objective-C developer.](#) See more — or post a gig for free! — on the [Jobs Board](#).

Avoiding iPhone App Rejection From Apple

by [Brian Stormont](#) on April 15, 2009 · [39 comments](#)

Guest author [Brian Stormont \(Profile\)](#) runs [Stormy Productions](#). Stormy Productions has created more than 45 iPhone apps that have been downloaded over 400,000 times!



I've been developing apps for the iPhone for over 6 months now. Over this time period, I've successfully submitted over 45 apps, the majority under [my own company's iTunes account](#). Given the large number of app submissions, I've had my share of app rejections.

As has been mentioned many many times on the various developer forums, Apple's approval process can be very frustrating and inconsistent. However, if you are careful, you can greatly reduce your risk of getting an app rejected.

App Rejected

Why an iPhone app may get rejected.

SEARCH

26
May
2009

No publicly available infrastructure to support diagnostic analysis

Category: [No publicly available infrastructure to support diagnostic analysis](#)

We've reviewed and determined that we cannot post this version of your iPhone application to the App Store because it provides to the user potentially inaccurate diagnostic functionality for iPhone OS devices. There is currently no publicly available infrastructure to support diagnostic analysis. This may result in your app reporting potentially inaccurate [...]

[Continue reading »](#) [1 Comment](#)

22
May
2009

References a pre-release version of iPhone OS/SDK

Category: [SDK pre-release version](#)

Thank you for submitting your app xxxx 1.0 to the App Store. We've reviewed and determined that we cannot post it because it references a pre-release version of iPhone OS/SDK. Referencing a pre-release version of the OS/SDK is in violation of Section 2.1 of the iPhone SDK Agreement: "...Apple may provide You with pre-release [...]"

[Continue reading »](#) [No comments](#)

CATEGORIES

- › [App Description Issue](#)
- › [App doesn't work as advertised](#)
- › [Apple Trademark Images](#)
- › [Application Versioning](#)
- › [Excessive Volumes of Data](#)
- › [Human Interface Guidelines](#)
- › [Lite Versions](#)
- › [Minimal user functionality](#)
- › [No publicly available infrastructure to support diagnostic analysis](#)
- › [SDK pre-release version](#)
- › [Section 3.3.1](#)
- › [Section 3.3.15](#)
- › [TableView Issues](#)
- › [Undocumented APIs](#)

RECENT POSTS

- [No publicly available infrastructure to support diagnostic analysis](#)
- [References a pre-release version of iPhone OS/SDK](#)
- [Minimal user functionality](#)
- [Undocumented APIs](#)

Application Submission Feedback

Unpublished rules and clarifications from Apple's App Review team that can cause your iPhone app to be rejected.

Are we missing one of your rejection reasons that other developers may not know about? [Submit it.](#)

[permalink](#)

KEYWORDS CANNOT CONTAIN NAMES OF OTHER APPS

Submitted by Dan Fabulich:

We've reviewed [app] and determined that we cannot post this version of your iPhone application to the App Store at this time because of inappropriate 'Keywords' used to identify your application. We will not post applications that reference other applications in their search criteria. It would be appropriate to remove [name of a competitor's app]."

Keyword terms must be related to your application content and cannot contain offensive terms. It is not appropriate to reference other applications.

This is great news. One popular sales-inflating technique we've all seen is to list popular or competing apps in the description field to be included in search queries for them, e.g. "Perfect for fans of Flight Control, Koi Pond, iFart Mobile, and iShoot!" This prohibits such techniques in the Keywords.

Note that this rejection only cites inclusion of such words in the (relatively new) Keywords field. We do not know whether the policy applies, or will be applied, to the Description field.

Go

TOP TAGS

App Store	14	conference	13	Objective-C	12	Switzerland	13
Apple	12	iPad	12	open source	10	Video	12
Business	12	iPhone	61	software	12	All Tags	175

AUG 3, '09
7:45 AM

Risk Management in iPhone Projects

AUTHOR **Adrian Kosmaczewski**

CATEGORIES **Opinion, iPhone**

TAGS **Apple, attorney, consulting, contract, iPhone, lawyer, legal, Opinion, technology**

[EDIT](#)

Let's be frank: it's not the best time to be an iPhone developer right now. In just one year of existence, the App Store seems to have evolved from the **hottest** to the **lamest** status, without any time to breathe in the middle, but with some **warning signs** every so often.



Several iPhone developers have publicly stated their opposition to the **Google Voice fiasco** (starting with Riverturn themselves, the developers of the application), and many have simply stopped creating iPhone OS applications altogether; just to name a few, **Fraser Speirs**, **Steven Frank** and **Andrew Wulf** have publicly stated that they don't want to deal with the App Store process anymore. And I'm sure that there are many more developers evaluating this very

CATEGORIES

Select Category

CONTACT

From 0800 to 1800 CET
Tel: + 41 78 739 47 76
Mail: software@akosma.com
AIM: [akosma@mac.com](aim:akosma@mac.com)
Skype: **akosma**
Twitter: **@akosma**

TWITTER

Going to Geneva. Time for Dev Day for iPhone

<http://akos.ma/t4y>

wooooot!

13 mins ago

[More updates...](#)

SHARE

[del.icio.us](#)

[Digg](#)

[Evernote](#)

[Facebook](#)

[FriendFeed](#)

[Google](#)

PARTNERS

Coriolis Technologies

Creates custom software for the iPhone and for the Apple Macintosh, in C/C++ or Cocoa.

Devego

They can make you an awesome website. They are devoted to create websites that are stunning and easy to use.

Infobahn

Infobahn creates and deploys internet, extranet and intranet sites using their own content management system EZMODIFIER™

Trifork A/S

Trifork develops, sells and delivers IT solutions to the financial and public sector in Denmark. Trifork A/S has 130 employees distributed in offices in Aarhus, Copenhagen, California and Switzerland.

Zerofee

Ethical design agency – they create visual identity and design for print and digital media, but not for irresponsible brands or companies.

Your Objective:

avoid this



and this



<http://smokingapples.com/iphone/app-store-iphone/the-worst-twitter-client-ever/>



I can't find one redeeming quality about this app. It's slow to start [on a 3GS], doesn't respond to taps while it's trying to load other things, and **crashes if you try to change modes a lot**. It's limited to only timeline, replies, and messages. It has no other functionality. Oh wait... I forgot its killer feature, you can have custom backgrounds and choose the color of your tweets. That **totally makes up for its lack of useful features and sluggish performance**. I'm not sure why someone would bother building such an inferior app other than that they wanted to find some suckers and score a quick buck. It seems even more insane to me that they'd be actively seeking out reviewers to cover this. I was given a promo code for ChillTwit, and **even for free I didn't want it on my phone**. I was sad just from looking at screenshots. Actually seeing it running confirmed all of my fears. If it was a free app, I might forgive the developer, but **the fact that he's trying to get \$0.99 out of people pisses me off to no end**. Go buy Tweetie.

If you somehow weren't scared away by all my bitching and whinning, you can see ChillTwit on the app store [here](#). But seriously, **if you buy this, we're not friends anymore**.



4

**Thou shalt optimize for
performance**

- Drawing and scrolling
- Application launch
- Files and data
- Power and battery life

Drawing and scrolling

- UIView subclasses are already optimized
- Custom views should use `setNeedsDisplayInRect`: whenever possible
- Cache static objects

- Use opaque views
- Avoid allocating while scrolling
- Reuse table cells
- Collapse view hierarchies

Application Launch

- Design apps for **quick launch** and **short use**
- Load data lazily
- Load only images needed

Files and Data

- Use Core Data for large datasets
- Avoid loading large files in memory
- Use plist files for structured static data

Power management

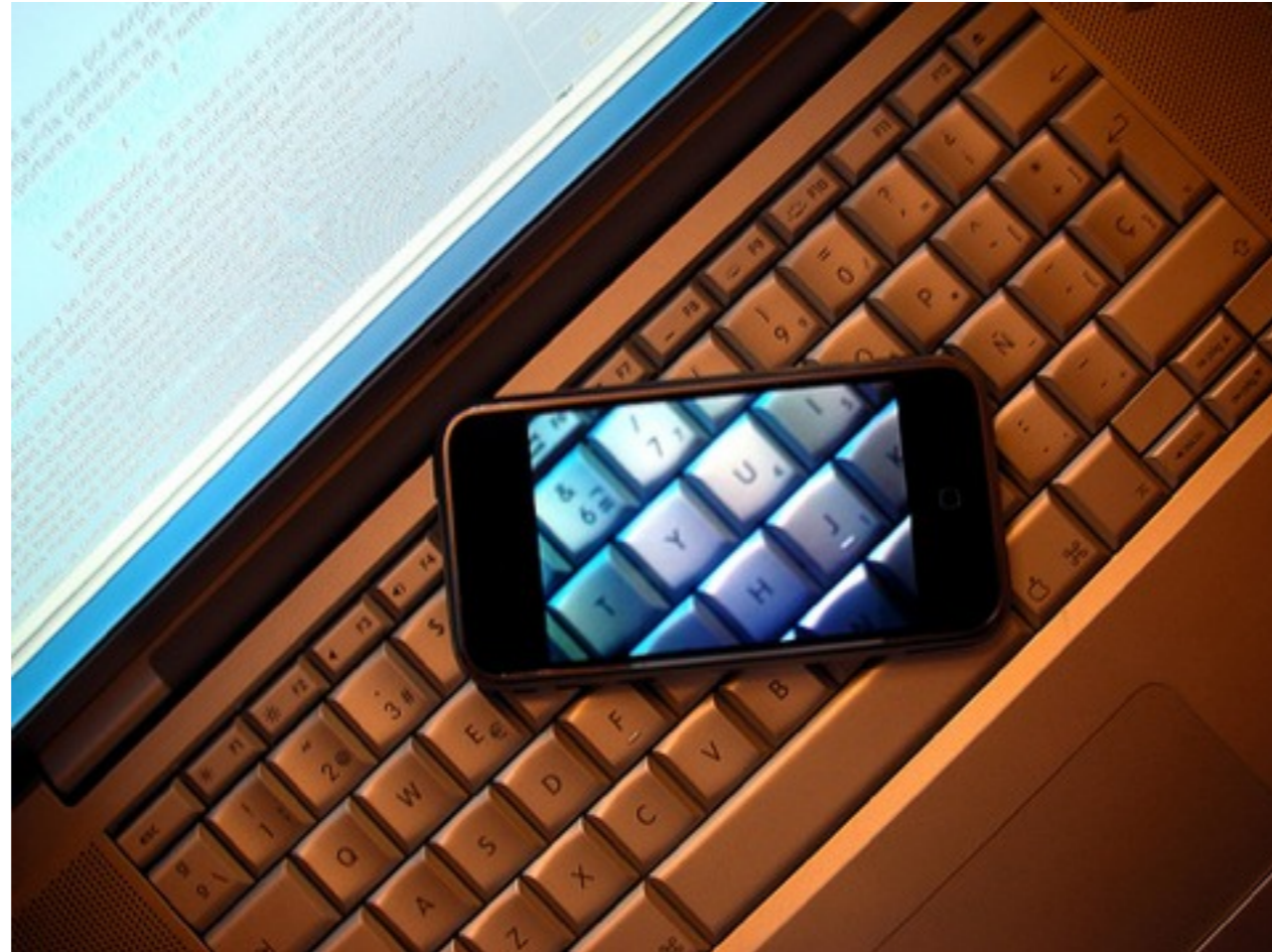
- 3G communications are expensive
 - Wi-Fi slightly cheaper
- Send small chunks of data at low frequency
- Prefer “chunky” to “chatty” protocols
- Better performance ==
longer battery life

5

**Thou shalt test in the
device**



<http://blogs.tech-recipes.com/itouchmyiphone/2008/03/26/from-iphone-sdk-to-simple-app-in-less-than-452-seconds/>



<http://www.flickr.com/photos/edans/1526393678/>

- Camera
- Accelerometer
- GPS
- Compass
- Battery
- Network
- ... Speed!





<http://www.flickr.com/photos/schill/969088410/>



<http://www.flickr.com/photos/jaytamboli/3788327603/>

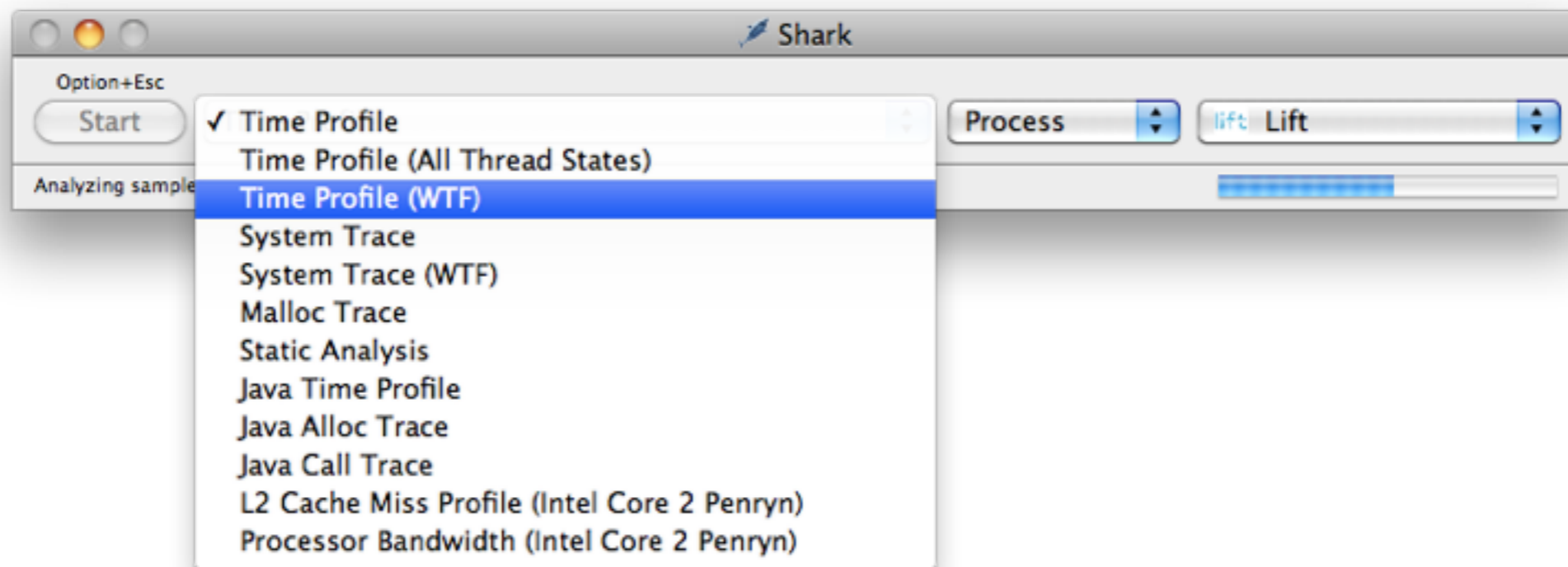
**Keep your old 3G(S)
or iPod touches!**

6

**Remember your
developer tools**



Shark



Session 3 - Time Profile of Lift

Profile Chart

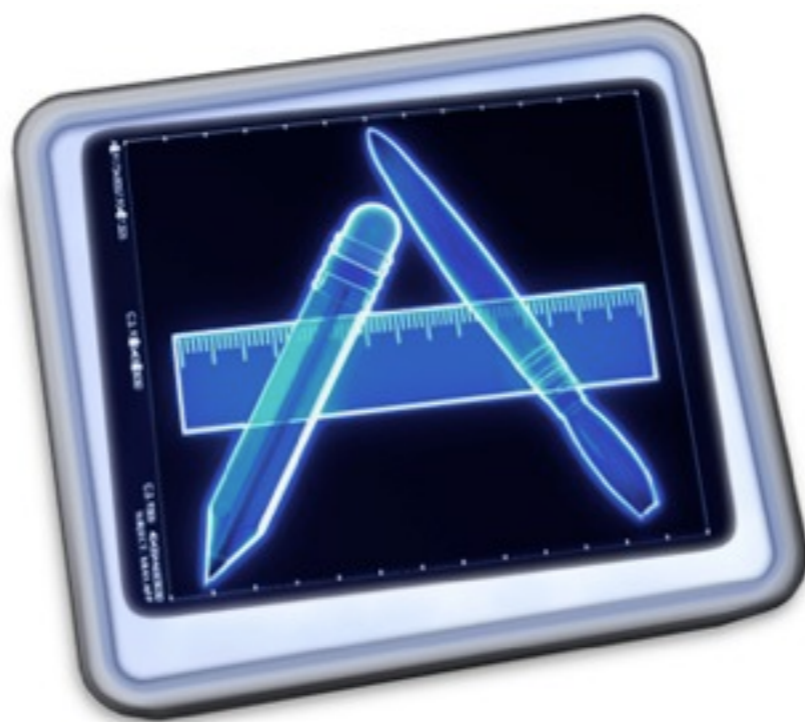
Tree (Top-Down)

!	Self	Total	Library	Symbol
	0.0%	57.1%	Lift	▼ start
	0.0%	57.1%	Lift	▼ main
	0.0%	57.1%	UIKit	▼ UIApplicationMain
	0.0%	57.1%	GraphicsServices	▼ GSEventRun
	0.0%	57.1%	GraphicsServices	▼ GSEventRunModal
	0.0%	57.1%	CoreFoundation	▼ CFRunLoopRunInMode
	0.0%	57.1%	CoreFoundation	▼ CFRunLoopRunSpecific
	0.0%	27.0%	Foundation	▼ __NSThreadPerformPerform
	0.0%	25.8%	Celestial	▼ -[AVObjectRegistry safeInvokeWithDescripti
	0.0%	23.9%	Celestial	▼ -[AVFileValidator fileCheckResult:]
	0.0%	23.3%	Foundation	▼ -[NSNotificationCenter postNotification
	0.0%	23.3%	CoreFoundation	▼ _CFXNotificationPostNotification
	0.0%	23.3%	Foundation	▼ _nsnote_callback
	0.0%	23.3%	MediaPlayer	▼ -[MPMoviePlayerController _create
	0.0%	19.0%	MediaPlayer	▼ -[MPAVController _avController]
	0.0%	18.4%	MediaPlayer	▼ -[MPAVController _connectAVC
	0.0%	18.4%	Celestial	▼ -[AVController init]
	0.0%	18.4%	Celestial	▼ -[AVController initWithQue
	0.0%	18.4%	Celestial	▼ -[AVController initWithQ
	0.0%	18.4%	MediaToolbox	▼ FigSharedPlayerCreate
	0.0%	18.4%	MediaToolbox	▼ FigPlayerAsyncCreate
	0.0%	18.4%	CoreMedia	▼ FigThreadCreate
	0.0%	18.4%	CoreMedia	▼ figSetMachPrioriti
	0.0%	18.4%	libSystem.B.dylib	▼ nthread_mach

No samples selected, 163 of 665 (24.5%) samples displayed

Process: (100.0%) Lift [86309] Thread: (24.5%) 0x7E247A8 View: Tree (Top-Down)

Instruments



	Run	⌘⇧↵
	Run – Breakpoints Off	⌘⇧R
	Debug – Breakpoints On	⌘⇧Y
	Run with Performance Tool	▶
	Stop	⌘⇧↵
	Attach to Process	▶
	Debugger	⌘⇧Y
	Mini Debugger	⌘⇧⌘
	Console	⌘⇧R
	Clear Console	⌘⇧⌘
	Show	▶
	Debugger Display	▶
	Variables View	▶
	Activate Breakpoints	⌘⇧\
	Stop on Objective-C Exceptions	
	Manage Breakpoints	▶
	Fix	
	Pause	⌘⇧P
	Step Into	⌘⇧I
	Step Over	⌘⇧O
	Step Out	⌘⇧T
	Next Thread	⌘⇧↑
	Previous Thread	⌘⇧↓
	Sync with Debugger	
	✓ Stop on Debugger()/DebugStr()	
	Enable Guard Malloc	

Instruments
Zombies
UI Recorder
Time Profiler
Threads
Sudden Termination
Multicore
GC Monitor
File Activity
Dispatch
Core Data
System Usage
OpenGL ES
Core Animation
Object Allocations
Leaks
CPU Sampler
Activity Monitor
Shark

Instruments

00:00:43 Run 1 of 1

Record Lift Default Target Flags Inspection Range Mini View Library

Instruments

ObjectAlloc

Leaks

Leaks Discovered

Total Leaked Bytes

Extended Detail

dyld

dyld::load(char const*, dyld::LoadCont...

dyld

dyld::libraryLocator(char const*, bool, ...

dyld

ImageLoader::recursiveLoadLibraries(l...

dyld

ImageLoader::link(ImageLoader::LinkC...

dyld

dyld::link(ImageLoader*, bool, ImageLo...

dyld

dlopen

dyld

0x95ab0f48

0x30288c69

_FillNextTokenInCache

DataDetectorsCore

DDTokenCacheGetTokensAtPosition

DataDetectorsCore

DDTokenCacheGetTokenOrSubtokenAf...

DataDetectorsCore

DDScannerScanQuery

DataDetectorsCore

-[DDOperation main]

DataDetectorsUI

0x30577345

0x3050a79d

0x3050a338

0x95ad7f39

0x95ad7dbe

Leaks - Lift

Leaked Object	#	Address	Size	Responsible Library
NSCFString	561	< multiple >	17.30 KB	
NSCFString	116	< multiple >	2.34 KB	
Malloc 128 Bytes		0x42698a0	128 Bytes	DataDetectorsCore
Malloc 128 Bytes		0x420eed0	128 Bytes	dyld
Malloc 128 Bytes		0x420a0d0	128 Bytes	dyld
Malloc 128 Bytes		0x1765380	128 Bytes	DataDetectorsCore
Malloc 128 Bytes		0x173adc0	128 Bytes	dyld
Malloc 128 Bytes		0x173ace0	128 Bytes	dyld
Malloc 128 Bytes		0x17255a0	128 Bytes	dyld

Leaks Configuration

Automatic Leaks Checking

Gather Leaked Memory Contents

Sampling Options

sec Between Auto Detections: 10.0

Leaks Status

Auto-Leaks: Idle

Check Manually

Check for Leaks Now

Grouping

Individual Leaks

Identical Backtraces

Call Tree

Invert Call Tree

Hide Missing Symbols

Hide System Libraries

Show Obj-C Only

Flatten Recursion

Call Tree Constraints

Specific Data Mining

Leaked Blocks

Q- All Fields

7

**Thou shalt use PNG
files**

Portable Network Graphics (PNG) is a bitmapped image format that employs lossless data compression. PNG was created to improve upon and replace GIF (Graphics Interchange Format) as an image-file format not requiring a patent license. It is pronounced /'pɪŋ/ [1] or spelled out as P-N-G. The PNG acronym is optionally recursive, unofficially standing for “**P**NG's **N**ot **G**IF”. [2]

Advantages

Compression on build

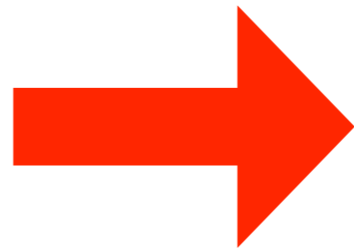
Low memory footprint

**Supported by all
editors**

Quality



Transparency

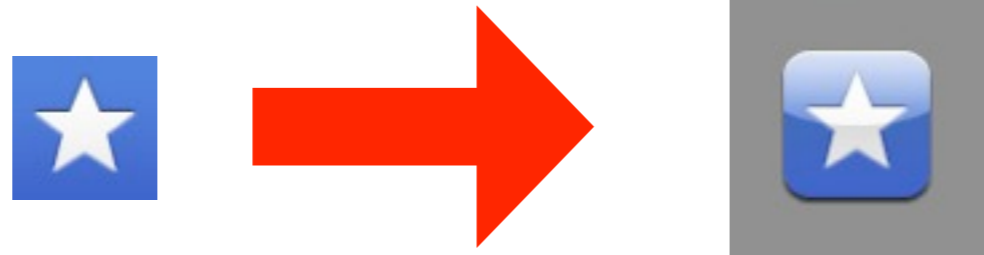


- App icon: 57x57
“Icon.png”
- Default image: 480x320
“Default.png”
- Settings icon: 29x29
“Icon-Small.png”

App Icon

**Design in maximum
quality, then reduce**

Do not apply effects



**512x512 72 DPI TIFF
file
for App Store**

8

**Thou shalt use static
analysis**

The LLVM Compiler Infrastructure

Site Map:

[Overview](#)
[Features](#)
[Documentation](#)
[Command Guide](#)
[Support](#)
[FAQ](#)
[Publications](#)
[LLVM Projects](#)
[Open Projects](#)
[LLVM Users](#)
[LLVM Developers](#)
[Bug Database](#)
[LLVM Logo](#)
[Wiki](#)

Download!

Download now:
[LLVM 2.5](#)

Try the
[online demo](#)

View the open-source
[license](#)

Search this Site

LLVM Overview

Low Level Virtual Machine (LLVM) is:

1. [A compilation strategy](#) designed to enable effective program optimization across the entire lifetime of a program. LLVM supports effective optimization at compile time, link-time (particularly interprocedural), run-time and offline (i.e., after software is installed), while remaining transparent to developers and maintaining compatibility with existing build scripts.
2. [A virtual instruction set](#) - LLVM is a low-level object code representation that uses simple RISC-like instructions, but provides rich, language-independent, type information and dataflow (SSA) information about operands. This combination enables sophisticated transformations on object code, while remaining light-weight enough to be attached to the executable. This combination is key to allowing link-time, run-time, and offline transformations.
3. [A compiler infrastructure](#) - LLVM is also a collection of source code that implements the language and compilation strategy. The primary components of the LLVM infrastructure are a GCC-based [C & C++ front-end](#), a link-time optimization framework with a growing set of global and interprocedural analyses and transformations, static back-ends for [many popular \(and some obscure\)](#) architectures, a back-end which emits portable C code, and a Just-In-Time compilers for several architectures.

Latest LLVM Release!

March 2, 2009: LLVM 2.5 is now [available for download!](#) LLVM is publicly available under an open source [License](#). Also, you might want to check out [the new features](#) in SVN that will appear in the next LLVM release. If you want them early, [download LLVM](#) through anonymous SVN.

Upcoming Releases

LLVM 2.6 release schedule:

- Aug 21 - Code Freeze
- Aug 28 - Pre-release1 testing begins
- Sept 04 - Pre-release1 testing ends
- TBD - Pre-release2 testing begins
- TBD - Pre-release2 testing ends
- TBD - Release!

[LLVM Home](#)
[Clang Home](#)

Events

[October 2, 2009 - LLVM/Clang
Developers' Meeting](#)

Quick Links

[About the Analyzer](#)
[Filing Bugs](#)

User Manual

[Obtaining the Analyzer](#)
[Running the Analyzer](#)
[Available Checks](#)
[Source-level Annotations](#)

Clang Mailing Lists

[cfe-dev](#)
[cfe-commits](#)



Clang Static Analyzer

The Clang Static Analyzer consists of both a source code analysis framework and a standalone tool that finds bugs in C and Objective-C programs. The standalone tool is invoked from the command-line, and is intended to run in tandem with a build of a project or code base.

Both are 100% open source and are part of the [Clang](#) project.

Download

Mac OS X

- Latest build (Universal binary, 10.5+): [checker-0.223.tar.bz2](#) (built September 29, 2009)
- [Installation](#) and [usage](#)

Other Platforms

For other platforms, please follow the instructions for [building the analyzer](#) from source code.

What is Static Analysis?

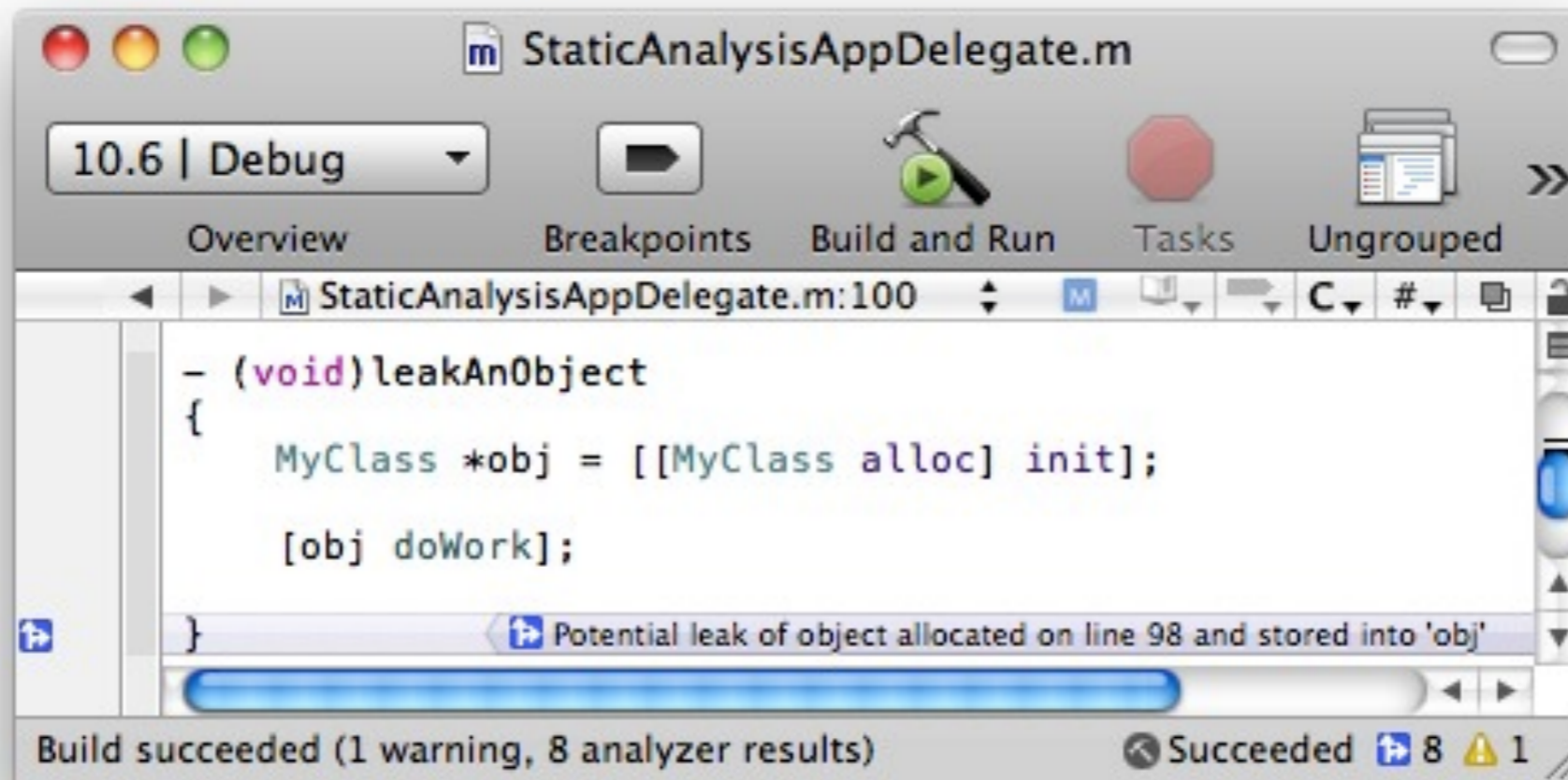
The term "static analysis" is conflated, but here we use it to mean a collection of algorithms and techniques used to analyze source code in order to automatically find bugs. The idea is similar in spirit to compiler warnings (which can be useful for finding coding errors) but to take that idea a step further and find bugs that are traditionally found using run-time debugging techniques such as testing.

Static analysis bug-finding tools have evolved over the last several decades from basic syntactic checkers to those that find deep bugs by reasoning about the semantics of code. The goal of the

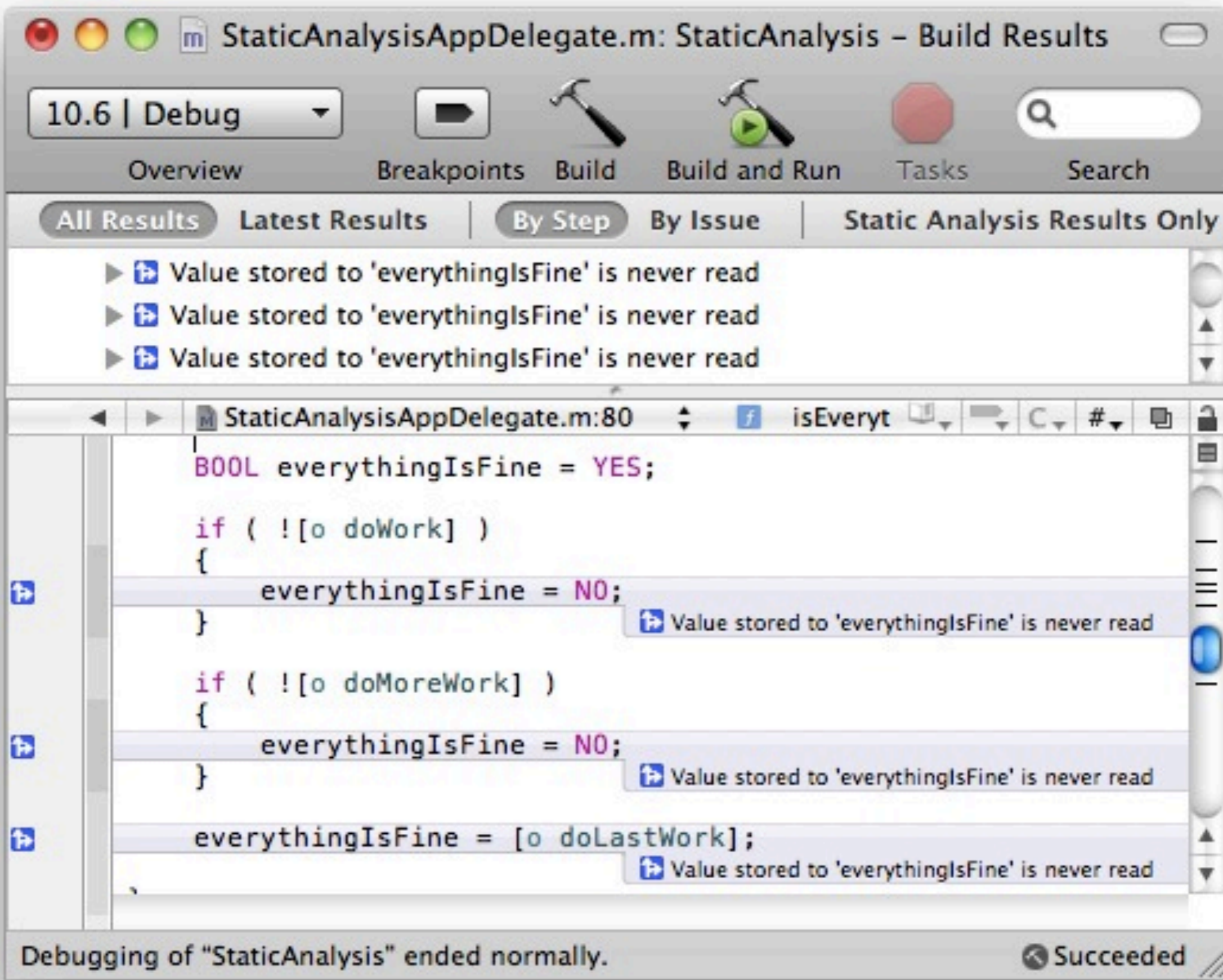
<http://clang-analyzer.llvm.org/>

Apple Xcode File Edit View Project Build Run Design SCM Window Help

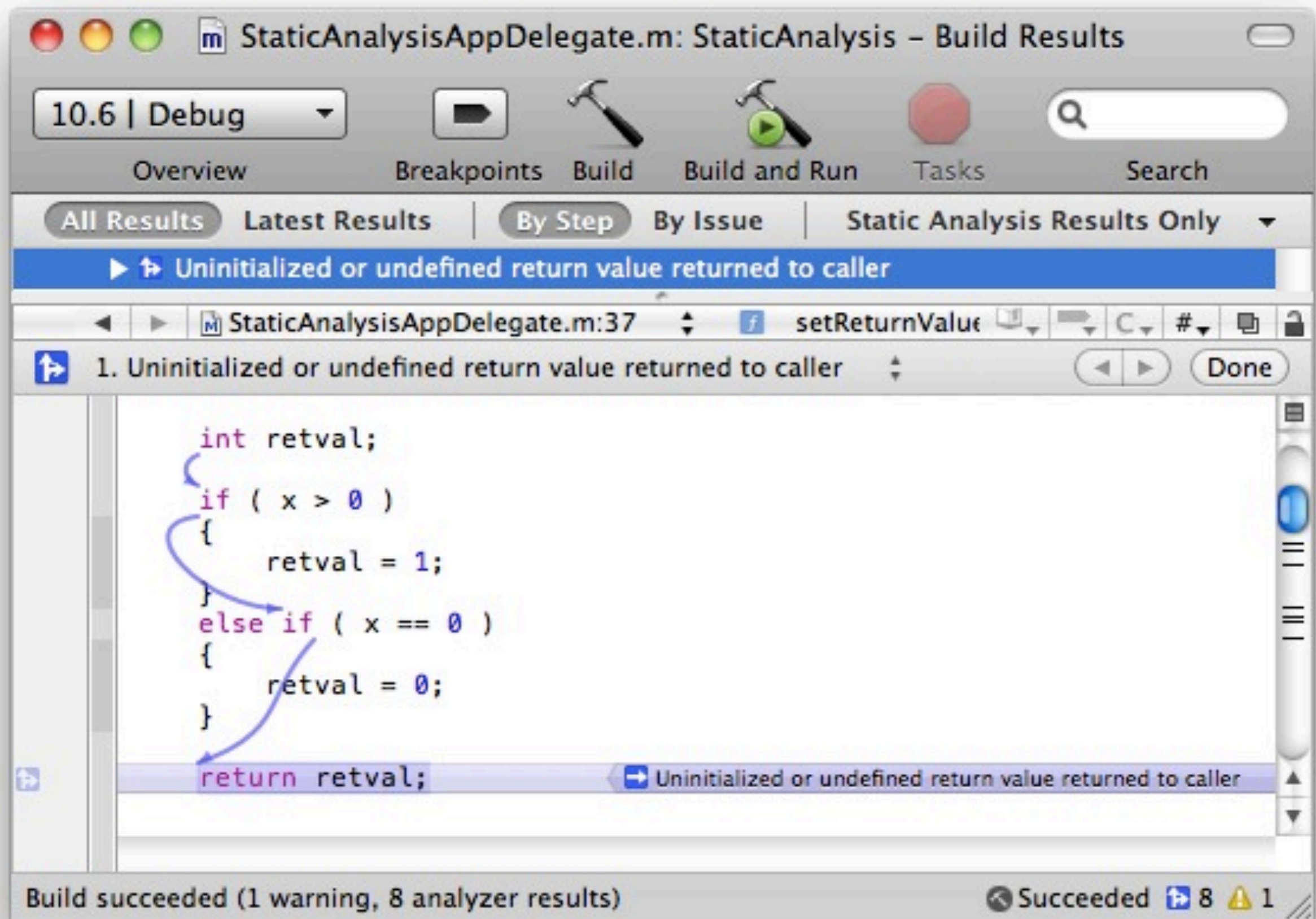
- Build Results ⌘⇧B
- Build ⌘B
- Build and Analyze** ⌘⇧A
- Build and Run ⌘↩
- Build and Run - Breakpoints Off ⌘R
- Build and Debug - Breakpoints On ⌘Y
- Clean ⌘⇧K
- Clean All Targets
- Next Build Warning or Error ⌘=
- Previous Build Warning or Error ⌘+
- Compile ⌘K
- Preprocess
- Show Assembly Code
- Touch



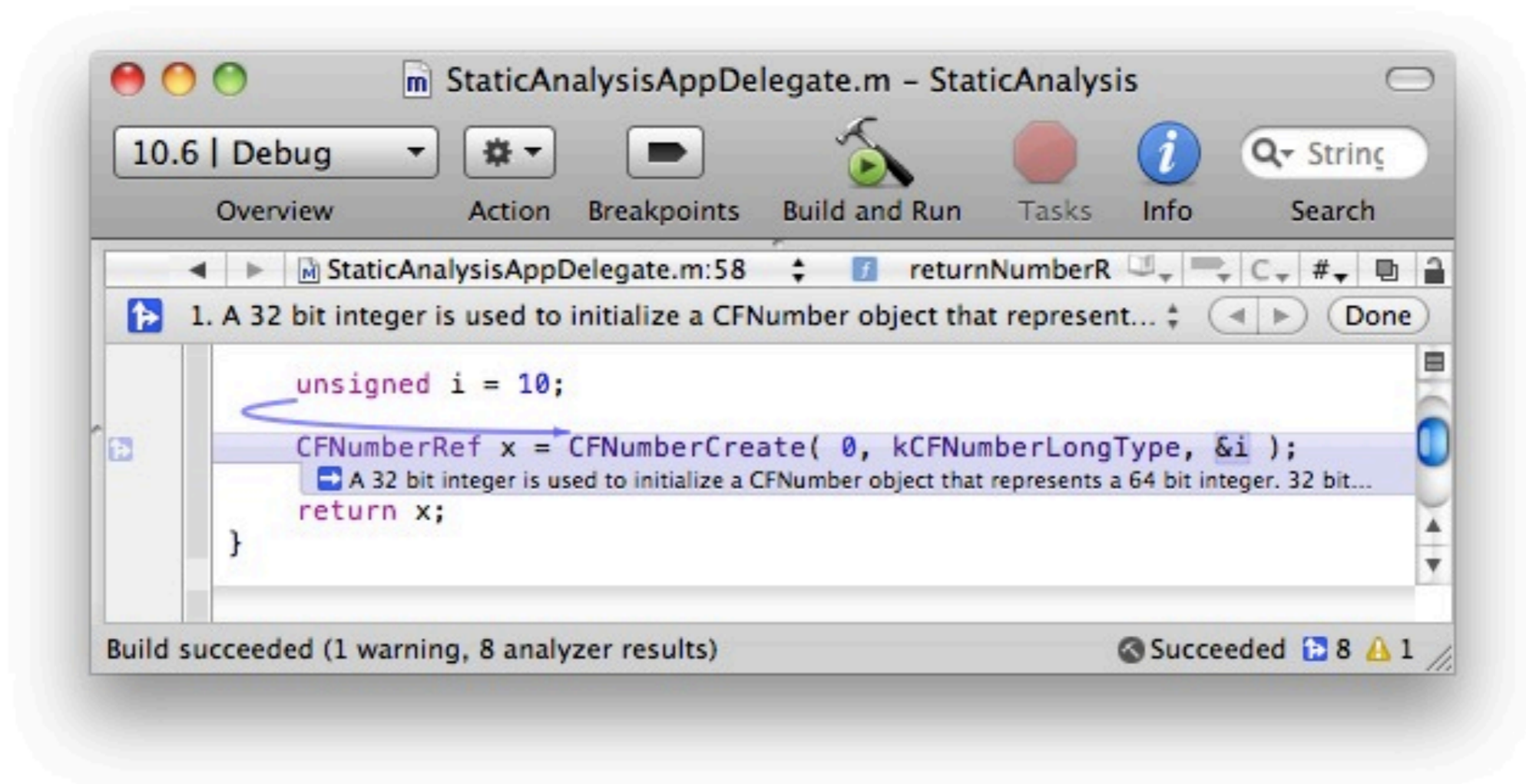
Source: Apple Documentation



Source: Apple Documentation



Source: Apple Documentation



Source: Apple Documentation

9

**Thou shalt have project
management hygiene**

- Project management
- Human resource management
- Developer working conditions
- Prototypes
- Quality management
- Code organization

TOP TAGS	App Store 14	conference 13	Objective-C 12	Switzerland 13
	Apple 12	iPad 12	open source 10	Video 12
	Business 12	iPhone 61	software 12	All Tags 175

JUL 28, '09 6:01 PM **Code Organization in Xcode Projects**

AUTHOR Adrian Kosmaczewski
CATEGORIES Tech, iPhone
TAGS Architecture, iPhone, Objective-C, Opinion, Tech, Xcode
[EDIT](#)

Xcode does not impose any structure to your source code tree. This is both cool and useful to quickly throw a couple of lines for a prototype, but in my experience, this approach does not scale. More often than not, without any hygiene, your project can become a mess. Just using Xcode defaults, after a while your resources will sit beside your .xcodeproj file, all the project classes will be thrown together in the Classes folder, and if you have a relatively large project, this approach makes finding individual files painful.

Of course, Xcode provides "Groups" to organize your source code, but the idea is to be able to quickly identify the different kind of files that make up your Xcode project, either for Mac or for the iPhone, without having to open the Xcode project file. This means having both a folder structure, and an internal source code file structure. All of this will help you maintain your project in the future, which means cheaper costs, and less time spent looking for bugs.

All of this is also particularly useful when browsing projects via Google Code, Github or any other kind of file view of source code repositories. If your code is organized in a nice folder structure, it is easier to explore than if all the files sit in the same folder.

CATEGORIES

CONTACT
From 0800 to 1800 CET
Tel: + 41 78 739 47 76
Mail: software at akosma.com
AIM: akosma@mac.com
Skype: akosma
Twitter: @akosma

TWITTER
Going to Geneva. Time for Dev Day for iPhone
<http://akos.ma/t4y>
wooooo!
18 mins ago
[More updates...](#)

SHARE
[del.icio.us](#)
[Digg](#)
[Evemote](#)
[Facebook](#)
[FriendFeed](#)
[Google](#)

PARTNERS
Coriolis Technologies
Creates custom software for the iPhone and for the Apple Macintosh, in C/C++ or Cocoa.
Devego
They can make you an awesome website. They are devoted to create websites that are stunning and easy to use.
Infobahn
Infobahn creates and deploys internet, extranet and intranet sites using their own content management system EZMODIFIER™
Trifork A/S
Trifork develops, sells and delivers IT solutions to the financial and public sector in Denmark. Trifork A/S has 130 employees distributed in offices in Aarhus, Copenhagen, California and Switzerland.
Zerofee
Ethical design agency – they create visual identity and design for print and digital media, but not for irresponsible brands or companies.

Your objective:

avoid chaos

NEW! Standish Group report shows more projects failing and less successful projects.

New CHAOS numbers show STARTLING RESULTS!

Boston, MA, - April 23 2009



Store



Sample Research

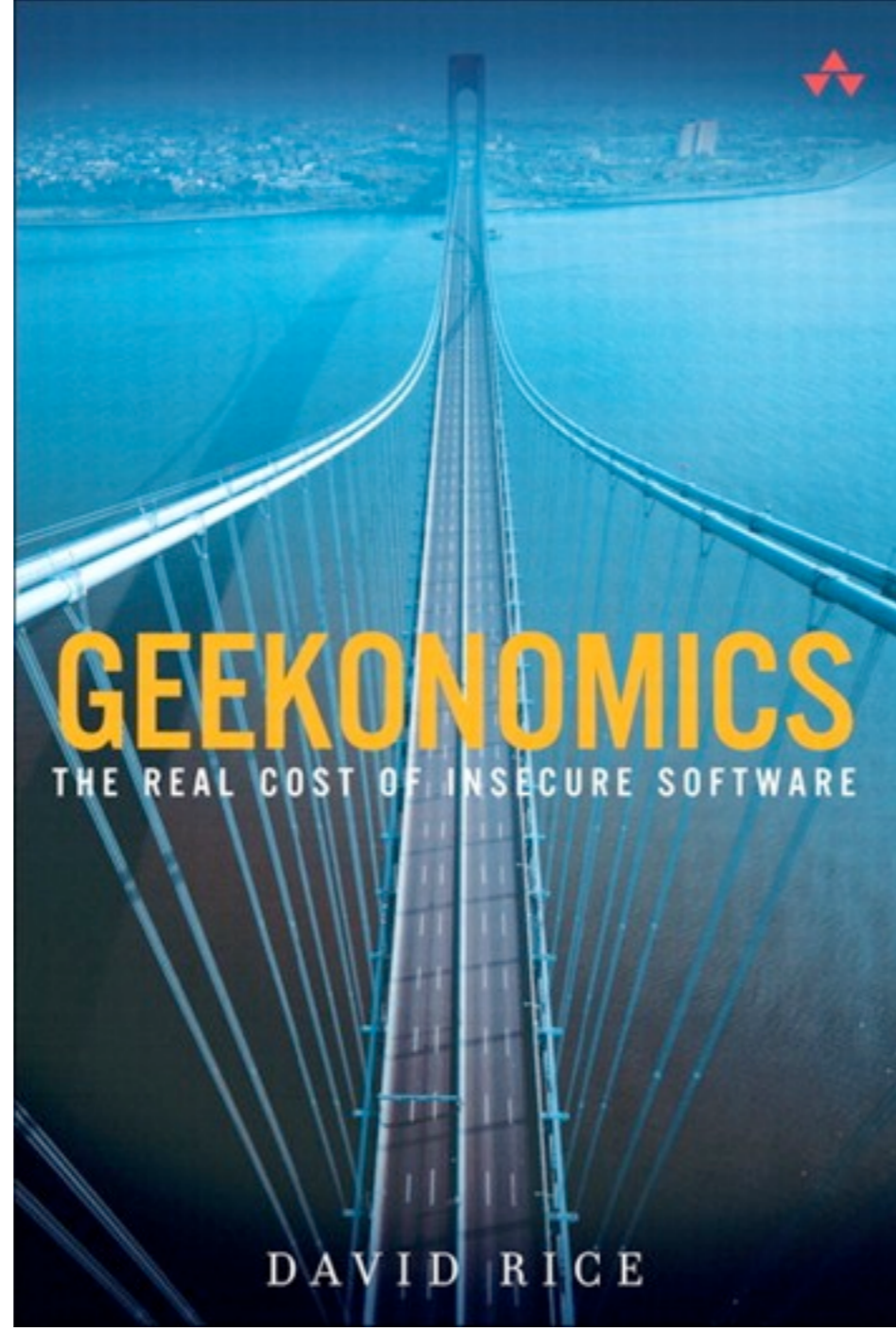


SURF

Boston, Massachusetts, April 23, 2009 - New Standish Group report shows more project failing and less successful projects.

The Standish Group's just-released report, "CHAOS Summary 2009," "This year's results show a marked decrease in project success rates, with 32% of all projects succeeding which are delivered on time, on budget, with required features and functions" says Jim Johnson, chairman of The Standish Group, "44% were challenged which are late, over budget, and/or with less than the required features and functions and 24% failed which

"These numbers represent a downtick in the success rates from the previous study, as well as a significant increase in the number of failures", says Jim Crear, Standish Group CIO, "They are low point in the last five study periods. This year's results represent the highest failure rate in over a decade"



<http://www.geekonomicsbook.com/>

10

Thou shalt have fun!

and

be creative!

Best iPhone Apps

The Guide for Discriminating Downloaders



O'REILLY®

Josh Clark



Recap'

1. Manage memory properly
2. Remove compiler warnings
3. Read the Human Interface Guidelines
4. Optimize for performance
5. Test in the device

6. Know your developer tools
7. Use PNG files
8. Use static analysis
9. Have project management hygiene
10. Have fun and be creative!

Thanks!

Slides available in
slideshare.net/akosma

Questions?

Ces slides sont disponibles avec une licence
**Creative Commons Attribution-No Derivative Works
3.0 Unported License**

<http://creativecommons.org/licenses/by-nd/3.0/>

