Future Proofing Your Application

Forewarned is forearmed

Henry Mason
iPhone Applications Engineer

Introduction

- Today’s apps should run on future iPhone OS versions
- We work hard to keep it that way
- We want your help

What You’ll Learn

- What causes binary compatibility problems
- What you can do to prevent them
- What is provided to help you

Non-Public APIs

- Private API usage nearly guarantees binary compatibility problems
- If it’s not in a header or documentation, do not use it
- If you need an API which is not available, ask us

Non-Public APIs

Even if it is in a header…

```
- (void)doSomethingSilly {
  UIViewController *myVC = self.myController;
  UINavigationController *myNC = self.navigationController;
  [myNC pushViewController:myVC animated:YES];
  // Do something after the myVC animation push is done
  [self performSelector:@selector(work) afterDelay:0.4];
}
```

✗

```
- (void)navigationController:(UINavigationController *)nc
didShowViewController:(UIViewController *)vc animated:(BOOL)animated {
  [self work];
}
```

✓

Non-Public APIs

Even if it is in a header…

```
- (void)grabImageOutOfEditor {
  UIVideoEditorController *myVEC = getVEC();
  NSArray *vcs = myVEC.viewControllers;
  UIViewController *vc = [vcs objectAtIndex:13];
  UIView *view = [vc.view.subviews objectAtIndex:42];
  CGImageRef img = view.layer.content;
  displayImage(img);
}
```

✗

Check out the new AVCaptureDevice API ✓

Categories Call for Care

Not a replacement for subclassing

```
@implementation UIView (MyAppAdditions)
- (void)drawRect:(CGRect)rect {
  [[UIColor greenColor] set];
  UIFillRect(rect);
}
@end
```

✗

```
@implementation MyView
- (void)drawRect:(CGRect)rect {
  [[UIColor greenColor] set];
  UIFillRect(rect);
}
@end
```

✓

Categories Call for Care

Prefix additions to system classes

```
@implementation UIView (MyAppAdditions)
- (CGFloat)height {
  return self.bounds.size.height;
}
@end
```

✗

```
@implementation UIView (MyAppAdditions)
- (CGFloat)myApp_height {
  return self.bounds.size.height
}
@end
```

✓

Categories Call for Care

Not just theoretical

```
@implementation NSView (MyAppAdditions)

- (BOOL)isHidden {
  // ...
}

@end
```

Framework Gotchas

Some APIs raise surprising exceptions

```
- (void)handleInput:(NSString *)userInput
{
  NSURL *url = [NSURL
  URLWithString:userInput];
  NSString *host = [url host];
  NSMutableString *mutableHost =
      [NSMutableString
      stringWithString:host];
  [mutableHost appendString:@".com"];
  [self handleHost:host];
}
```

✗

Framework Gotchas

Some APIs raise surprising exceptions

```
- (void)handleInput:(NSString *)userInput
{
  NSURL *url = [NSURL
  URLWithString:userInput];
  NSString *host = [url host];
  NSMutableString *mutableHost =
      [[host mutableCopy] autorelease];
  [mutableHost appendString:@”.com”];
  [self handleHost:host];
}
```

✓

Naming Conventions

Respect them

```
@interface UISpecialView : UIView

@property float specialness;

@end
```

✗

```
@interface MyAppSpecialView : UIView

@property float specialness;

@end
```

✓

UIKit Event Handling

- UITouch forwarding requires extreme caution
  - Every responder must be your custom subclass of UIView
  - Can’t be combined with other touch handling
- Remember -[UIResponder touchesCancelled:withEvent:]
- UIGestureRecognizer is your new best friend

UITableViewCell

Subviews are unusual

```
@implementation MyTableViewCell : UITableViewCell

- (id)initWithFrame:(CGRect)frame {
  if ((self = [super initWithFrame:frame])) {
    UIImage *img = [UIImage imageNamed:@”lol.jpg”];
    UIImageView *imgView = [[UIImageView alloc]
    initWithImage:img];
    [self addSubview:imgView];
    [imgView release];
  }
  return self;
}

@end
```

✗

UITableViewCell

Use UITableViewCell.contentView

```
@implementation MyTableViewCell : UITableViewCell

- (id)initWithFrame:(CGRect)frame {
  if ((self = [super initWithFrame:frame])) {
    UIImage *img = [UIImage imageNamed:@”lol.jpg”];
    UIImageView *imgView = [[UIImageView alloc]
    initWithImage:img];
    [self.contentView addSubview:imgView];
    [imgView release];
  }
  return self;
}

@end
```

✓

Don’t Block Your UI

The main thread is for your user

```
@implementation MyTableViewCell : UITableViewCell

- (void)startUpdateFromURL:(NSURL *)url {
  [self loadURL:URL];
}


- (void)loadURL:(NSURL *)url {
  NSString *name = [NSString stringWithContentsOfURL:url];
  UIImage *loadedImage = [UIImage imageNamed:name];
  self.imageView.image = loadedImage;
}

@end
```

Slow

UIKit: Usually Main Thread Only

```
@implementation MyTableViewCell : UITableViewCell

- (void)startUpdateFromURL:(NSURL *)url {
  [self performSelectorInBackground:@selector(loadURL:)
  withObject:URL];
}


- (void)loadURL:(NSURL *)url {
  // Runs on background thread
  NSString *name = [NSString stringWithContentsOfURL:url];
  UIImage *loadedImage = [UIImage imageNamed:name];
  self.imageView.image = loadedImage;
}

@end
```

❌

UIKit: Usually Main Thread Only

```
@implementation MyTableViewCell : UITableViewCell
- (void)startUpdateFromURL:(NSURL *)url {
  [self performSelectorInBackground:@selector(loadURL:)
  withObject:URL];
}

- (void)loadURL:(NSURL *)url {
  // Runs on background thread
  NSString *name = [NSString stringWithContentsOfURL:url];
  [self performSelectorOnMainThread:@selector(loadedName:)
  withObject:name];
}

- (void)loadedData:(NSString *)name {
  self.imageView.image = [UIImage imageNamed:name];
}
@end
```

✓

Now with Dispatch

```
@implementation MyTableViewCell : UITableViewCell

- (void)startUpdateFromURL:(NSURL *)url {
  [[[[NSOperationQueue alloc] init] autorelease]
  addOperationWithBlock:^{
    NSString *name = [NSString stringWithContentsOfURL:url];
    [[NSOperationQueue mainQueue] addOperationWithBlock:^{
      self.imageView.image = [UIImage imageNamed:name];
    }];
  }];
}

@end
```

Multitasking Considerations

-applicationDidEnterBackground: Must be fast

iOS 4

```
- (void)applicationDidEnterBackground:(UIApplication *)app {
  [self doLotsOfWorkWithObject:@"something"];
}
```

May prevent backgrounding

✗

```
- (void)applicationDidEnterBackground:(UIApplication *)app {
  UIBackgroundTaskIdentifier bti;
  bti = [app beginBackgroundTaskWithExpirationHandler:nil];
  if (bti != UIBackgroundTaskInvalid) {
    dispatch_async(dispatch_get_global(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0), ^{
      [self doLotsOfWorkWithObject:@"something"];
      [app endBackgroundTask:bti];
    });
  }
}
```

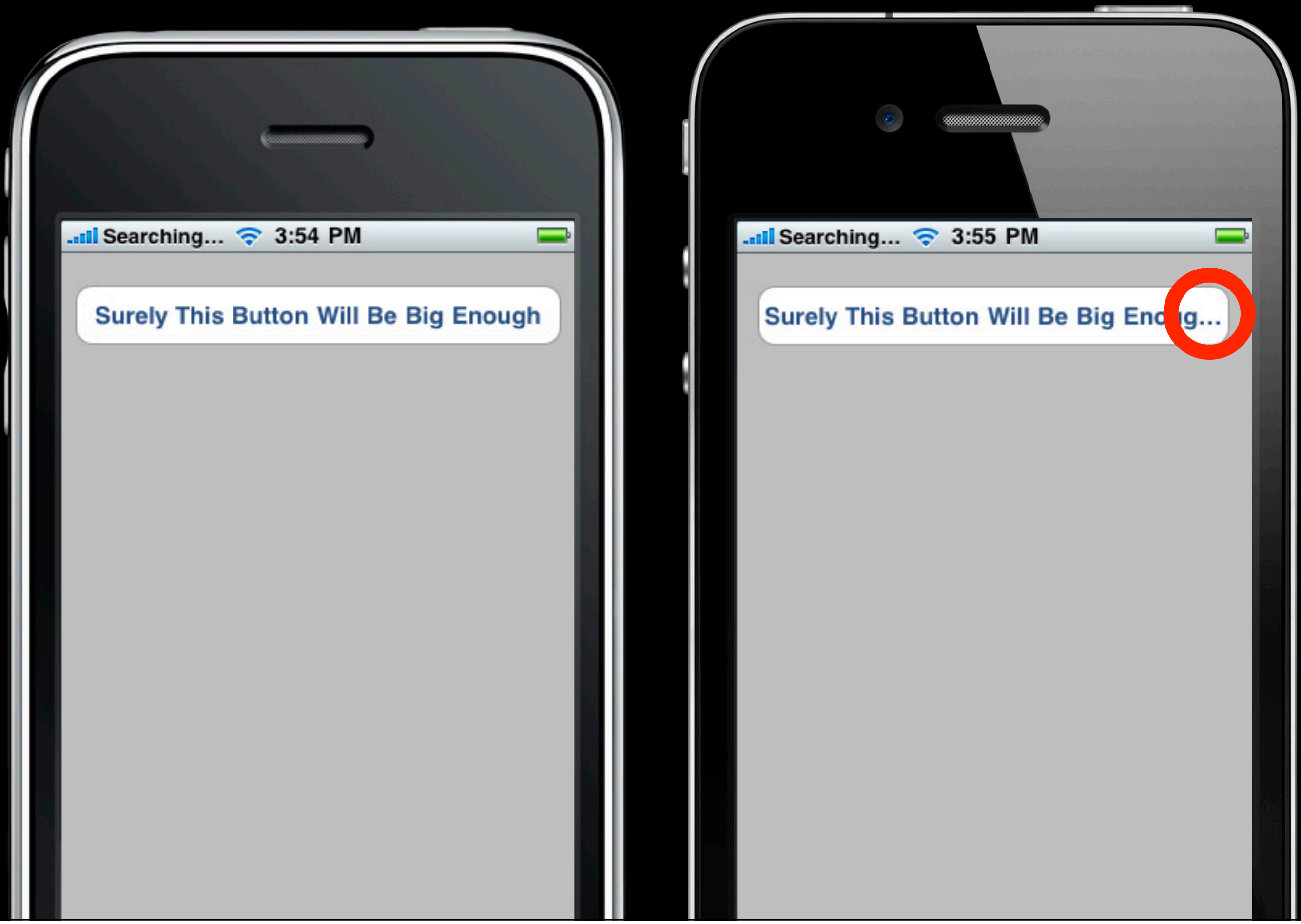
✓

Newly Thread-Safe in iOS 4

- UIImage (but not +imageNamed:)
- -[UIApplication beginBackgroundTaskWithExpirationHandler:]
- UIGraphics
  - Push/pop contexts
  - Fill and stroke rects
  - Draw UIImages
  - String drawing (e.g., -[NSString drawAtPoint:])

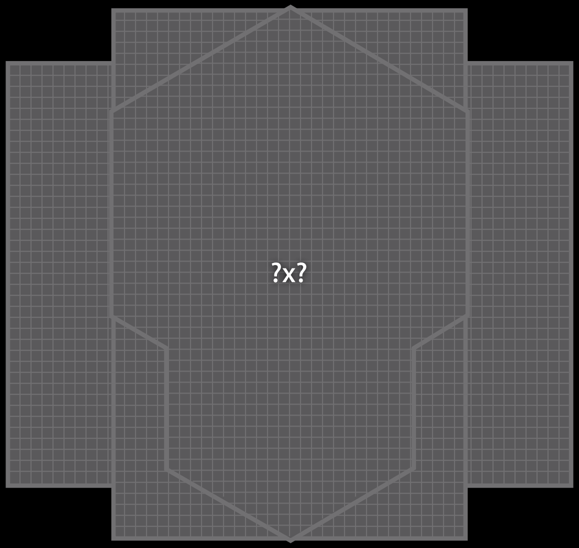
Assumptions

Make you look bad


Searching... 3:54 PM
Surely This Button Will Be Big Enough
Searching... 3:55 PM
Surely This Button Will Be Big Enough...

Pixels Are Implementation Details

- Font metrics and margins are not stable
- Critical for internationalization
- Brave new post-320x480 world



Do Use UIScreen.bounds

New screen sizes may be closer than they appear

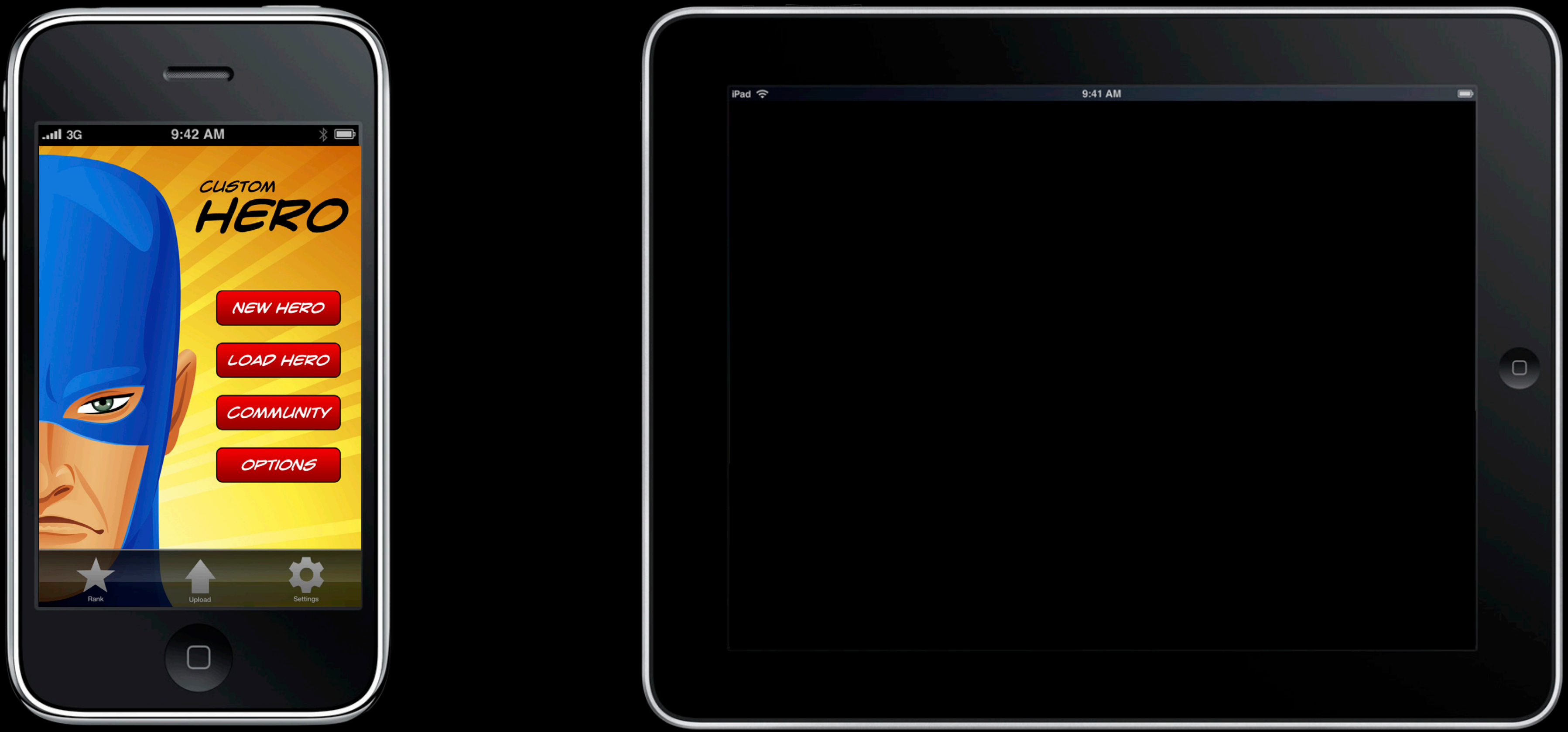
```
- (void)applicationDidFinishLaunching:(UIApplication *)app {
  UIViewController *root = self.rootViewController;
  root.frame = CGRectMake(0, 0, 320, 480);
  [self.window addSubView:root.view];
}
```

✗

```
- (void)applicationDidFinishLaunching:(UIApplication *)app {
  UIViewController *root = self.rootViewController;
  root.frame = [UIScreen mainScreen].bounds;
  [self.window addSubView:root.view];
}
```

✓

Where Did My App Go?



UIDevice.model

(Almost) never what you want

```
- (void)setupUserInterface {
  NSString *deviceModel = [UIDevice currentDevice].model;
  if ([deviceModel isEqualToString:@”iPhone”]) {
    [self setupMicrophoneInterface];
    [self setupCameraInterface];
    [self setupSMSInterface];
  } else if ([deviceModel isEqualToString:@”iPod Touch”]) {
    [self setupNoMicrophoneInterface];
    [self setupNoCameraInterface];
    [self setupNoSMSInterface];
  }
}
```

Use Capability-Based API

```
- (void)setupUserInterface {
  if ([AVAudioSession sharedInstance].inputIsAvailable) {
    [self setupMicrophoneInterface];
  } else {
    [self setupNoMicrophoneInterface];
  }
  if ([UIImagePickerController
  isSourceTypeAvailable:UIImagePickerControllerSourceTypeCamera]) {
    [self setupCameraInterface];
  } else {
    [self setupNoCameraInterface];
  }
  if ([[UIApplication sharedApplication] canOpenURL:[NSURL
  URLWithString:@"sms://123"]]) {
    [self setupSMSInterface];
  } else {
    [self setupNoSMSInterface];
  }
}
```

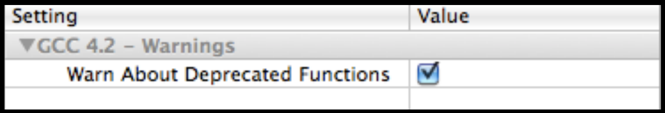
New Capability-Based API

- +[MFMessageComposeViewController canSendText:]
- EKCalendar.supportedEventAvailabilities
- Gyroscope: CMMotionManager
    - gyroAvailable
    - deviceMotionAvailable
- UIScreen.scale
- New camera availability API
    - +isCameraDeviceAvailable:
    - +isFlashAvailableForCameraDevice:
    - +availableCaptureModesForCameraDevice:

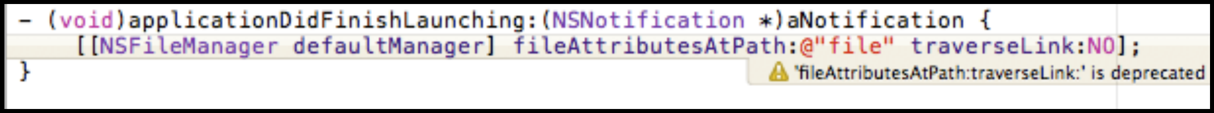
Tools

Do Cross a Bridge Before You Come to It

- Compile and test with beta SDKs ASAP
- Report unexpected behavior changes to us
- Pay attention to deprecation warnings
    - Means bugs in that API won't be fixed



Setting	Value
▼GCC 4.2 – Warnings	
Warn About Deprecated Functions	☑



```
- (void)applicationDidFinishLaunching:(NSNotification *)aNotification {
    [[NSFileManager defaultManager] fileAttributesAtPath:@"file" traverseLink:NO];
}
```

⚠ 'fileAttributesAtPath:traverseLink:' is deprecated

Linked-On-or-After

Nobody’s perfect

```
UIImage *myImage = [UIImage imageNamed:@"My Image"];
NSLog(@"[myImage retainCount] == %d", [myImage retainCount]);
```

iPhone OS 2.x:

```
[myImage retainCount] == 2
```

iPhone OS 3.0 and later, with 2.x apps:

```
[myImage retainCount] == 2
```

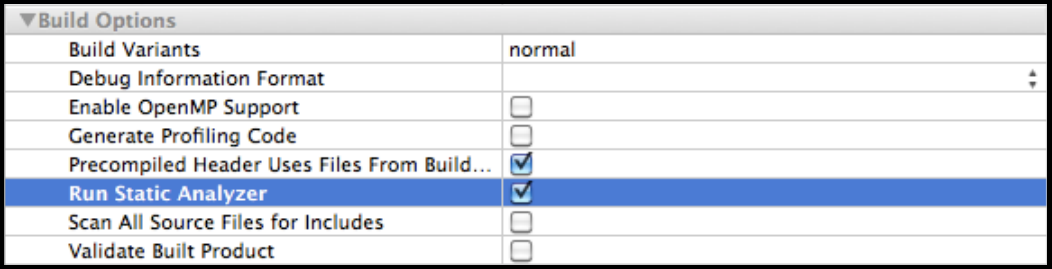
iPhone OS 3.0 later, with 3.0 and later apps:

```
[myImage retainCount] == 1
```

Static Analyzer

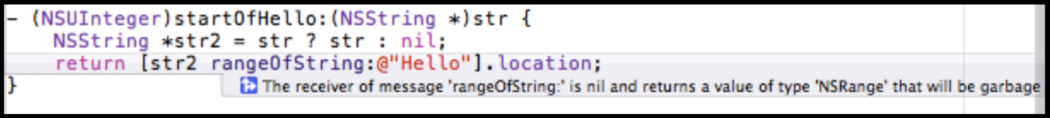
Catch bugs before they happen

- Catches tons of errors
  - …even those you don’t find at run time
- Turn it on by default



▼Build Options

Build Variants	normal
Debug Information Format	
Enable OpenMP Support	□
Generate Profiling Code	□
Precompiled Header Uses Files From Build…	☑
Run Static Analyzer	☑
Scan All Source Files for Includes	□
Validate Built Product	□



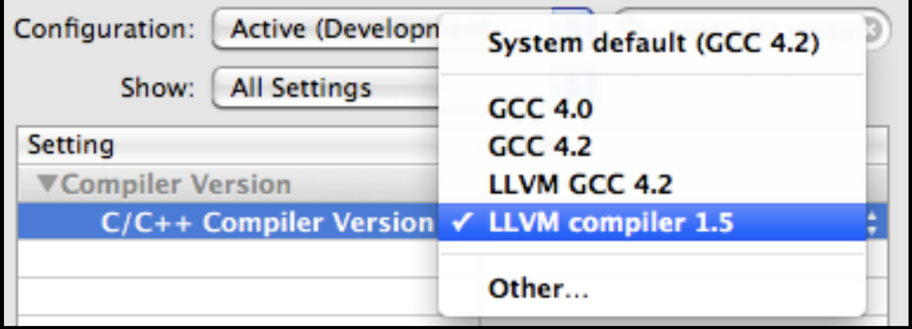
```
- (NSUInteger)startOfHello:(NSString *)str {
    NSString *str2 = str ? str : nil;
    return [str2 rangeOfString:@"Hello"].location;
}
```

The receiver of message 'rangeOfString:' is nil and returns a value of type 'NSRange' that will be garbage

Static Analyzer

One gotcha…

- No C++ or Objective-C++ (yet)
- Silently fails if LLVM can't compile it
- Catch by compiling with LLVM instead of GCC



Configuration: Active (Developn

Show: All Settings

Setting

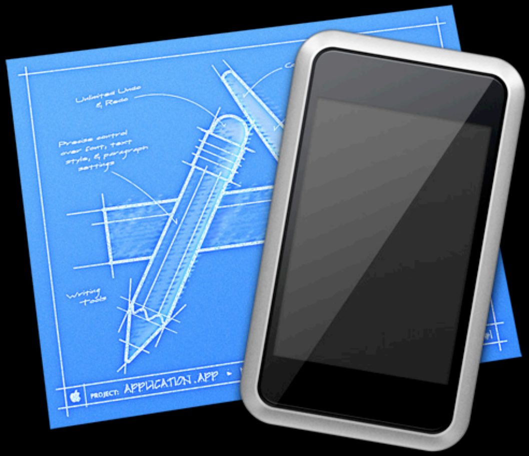
▼Compiler Version

C/C++ Compiler Version ✓ LLVM compiler 1.5

iPhone Simulator

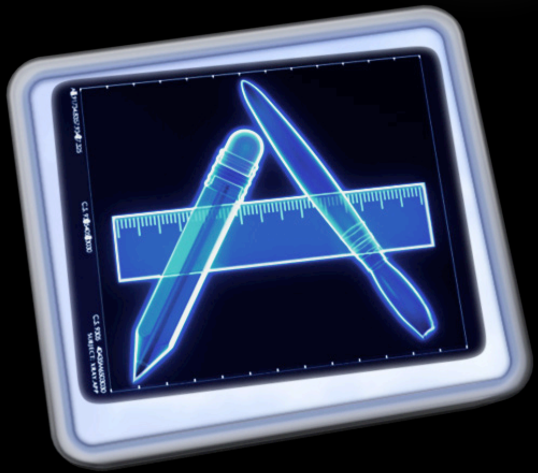
iOS 4

- New in iOS 4: Mix app and OS build versions
  - Particularly great for universal apps
- Test unusual configurations
  - In-call status bar
  - Low memory warnings
- Still not a replacement for devices
  - No new background modes
  - Different performance



Instruments

- Runtime problem detector
- Run *Leaks* and *Allocations* routinely
- New in iOS 4: Automation
  - Great for finding behavior changes with new OS versions



Calling New APIs on Old Devices

New C functions

```
- (void)doSomethingIfVoiceOverRunning {
  BOOL voiceOverRunning = NO;
  if (UIAccessibilityIsVoiceOverRunning != NULL) {
    voiceOverRunning = UIAccessibilityIsVoiceOverRunning();
  }
  [self handleVoiceOverRunning:voiceOverRunning];
}
```

Calling New APIs on Old Devices

New methods

```
- (void)doSomethingInBackgroundForApp:(UIApplication *)app {
  UIBackgroundTaskIdentifier BTI = UIBackgroundTaskInvalid;
  SEL sel = @selector
  (beginBackgroundTaskWithExpirationHandler:);
  if ([app respondsToSelector:sel]) {
    BTI = [app beginBackgroundTaskWithExpirationHandler:nil];
  }
  // Use BTI
}
```

Calling New APIs on Old Devices

New classes

```
- (void)doSomething {
  Class UINibClass = NSClassFromString(@"UINib");
  if (UINibClass) {
     UINib *nib = [UINibClass nibWithNibName:@"myNib"
  bundle:nil];
  } else {
    // Fallback behavior for Pre-4.0
  }
}
```

Calling New APIs on Old Devices

Subclassing new classes?

- Technically feasible with iPhone OS 3.1+ runtime
- Not possible with current compilers and linkers
- If you have a need for this, let us know

More Information

Bill Dudney
Frameworks Evangelist
dudney@apple.com

Documentation
Cocoa Fundamentals Guide
Introduction to Coding Guidelines for Cocoa
http://developer.apple.com

Apple Developer Forums
http://devforums.apple.com

Related Sessions

API Design for Cocoa and Cocoa Touch	Marina Thursday 4:30PM
Adopting Multitasking on iPhone OS, Part 1 (Repeat)	Marina Friday 9:00AM
Adopting Multitasking on iPhone OS, Part 2 (Repeat)	Marina Friday 10:15AM
What’s New in Cocoa Touch (Repeat)	Marina Friday 11:30AM
Mastering Xcode for iPhone OS Development, Part 1	Mission Tuesday 2:00PM
Mastering Xcode for iPhone OS Development, Part 2	Pacific Heights Wednesday 10:15AM

Labs

Application Compatibility Lab	Application Frameworks Lab B Thursday 4:30PM
Xcode 4 Lab	Developer Tools Lab B Friday 9:00AM

Summary

- Binary compatibility is hard, but together we've done pretty well
- Be proactive and vigilant about new OS versions
  - Don't hesitate to file bugs
- Take advantage of available tools

Q&A

WWDC10

[redacted]