

Introduction to iPhone Development



Introduction to iPhone Development

Contents		Task
1	Application Runtime Core Architecture and Life-cycles	<input type="checkbox"/>
2	What's in a bundle? The resources in an app bundle	<input type="checkbox"/>
3	Customizing Behavior How does it launch? IPC?	<input type="checkbox"/>
4	Relation with Tools How do we get the app on the phone?	<input type="checkbox"/>

Introduction to iPhone Development

Contents		Task
1	Application Runtime Core Architecture and Life-cycles	<input type="checkbox"/>
2	What's in a bundle? The resources in an app bundle	<input type="checkbox"/>
3	Customizing Behavior How does it launch? IPC?	<input type="checkbox"/>
4	Relation with Tools How do we get the app on the phone?	<input type="checkbox"/>

Application Life-Cycle

How your application lives and dies

User taps icon on home screen

main()

UIApplicationMain()



System requests termination

Application actually terminates

main and UIApplicationMain

- main()
 - Just like any other `main` functions (C, C++, etc)
 - Creates top-level autorelease pool
 - Starts application with `UIApplicationMain`
- UIApplicationMain()
 - Creates instance of `UIApplication` that is responsible for actually launching your application (loading main Nib file).
 - Takes four parameters: `argc`, `argv`, ignore other two.

Application Delegate

- Monitors high level or critical actions in application
 - Launch
 - Terminate
 - Memory warnings
- Conforms to Objective-C protocol (all methods optional)

The Main Nib File

- Remember: archive of objects.
- One of these objects is your main window.
- For now, think of the Main Nib file as your interface, as the year progresses, we'll show how to load additional Nib files.
- Interface elements *not* in your main Nib file:
 - Status bar
 - Application instance (we'll talk about proxy objects in three weeks)

Event-Handling Cycle

- System receives event (like mouse movement)
- System sends event on to your application instance
- Application instance then forwards the event to the First Responder, who starts sending the event up the chain. This is called the **responder chain**, and is a pretty advanced topic.
- Take-away: Event handling is complicated, and all you really need to know is that you can intercept these events to use them (like `touchesBegan:withEvent:`).

Life-Cycle Review

User touches icon on home screen
System calls main()
main() calls UIApplicationMain()
UIApplicationMain() creates instance of UIApplication
UIApplication instance loads main Nib file, sets up based on application properties
UIApplication instance goes into run loop, waiting for and forwarding events to interface elements (instances of UIResponder)
User taps home button or does another termination activity
UIApplication instance tells your delegate that the application is terminating
UIApplicationMain() exits, main() exits, process exits

Sandboxing

- iPhone OS does not give your application free reign like it does on OS X.
- Your application only has access to a certain part of the file system, something like: `/ApplicationRoot/ApplicationID`
- This is a security device preventing a single application from destroying your phone or iPod Touch.

Virtual Memory

- Virtual memory on the phone is quite interesting.
- It exists, but not really.
- It exists in that it gives your application the full virtual 32-bit address space.
- However, it does not write volatile pages to disk (b/c flash memory only has so many write cycles).
- Instead, the OS tries to free up nonvolatile memory.
- Therefore, you should (must) respond to `applicationDidReceiveMemoryWarning:` in your delegate

Introduction to iPhone Development

Contents		Task
1	Application Runtime Core Architecture and Life-cycles	<input checked="" type="checkbox"/>
2	What's in a bundle? The resources in an app bundle	<input type="checkbox"/>
3	Customizing Behavior How does it launch? IPC?	<input type="checkbox"/>
4	Relation with Tools How do we get the app on the phone?	<input type="checkbox"/>

Introduction to iPhone Development

Contents		Task
1	Application Runtime Core Architecture and Life-cycles	<input checked="" type="checkbox"/>
2	What's in a bundle? The resources in an app bundle	<input type="checkbox"/>
3	Customizing Behavior How does it launch? IPC?	<input type="checkbox"/>
4	Relation with Tools How do we get the app on the phone?	<input type="checkbox"/>

Application Contents

File	Description
MyApp	The actual application executable code
Icon.png	Your application's home screen icon
MainWindow.nib	The main Nib file containing your interface
Info.plist	Property list with information about your application
myimage.png	A non-localized image or other resource
Settings.bundle	Preference pane for the Settings app
Icon-Settings.png	Icon for settings application
Default.png	The image to show while your app is launching
en.lproj, fr.lproj, ...	Localized folders (ignore these for now)
Other items...	...that you should ignore for now.

MyApp

- The compiled, executable code for your application.
- Actual name is the name of your application bundle minus the “.app” extension.
- If your application bundle doesn't have this, you don't actually have an application, you just have a folder of stuff.

Property Lists

- Used for tons of data serialization in iPhone OS and OS X.
- XML or binary files.
- When loaded by Cocoa [Touch], the data in the property list gets turned into real objects.
 - `<string>Hello world!</string>` gets turned into an instance of `NSString` with the contents "Hello World!". Same applies for dictionaries (`NSDictionary`), arrays (`NSArray`), numbers (`NSNumber`), dates (`NSDate`), and data (`NSData`).
- Use property list editor in Xcode to edit property lists.





Info.plist

- Property list containing generic information as well as customized properties for your application.
- Required information:
 - Display name (`CFBundleDisplayName`)
 - Identifier (`CFBundleIdentifier`)
 - iPhone app? (`LSRequiresIPhoneOS`)
 - Main Nib file name (`NSMainNibFile`)
- Customized properties
 - Status bar style (`UIStatusBarStyle` or `UIStatusBarHidden`)
 - Others...





Info.plist and IPC

- You can do IPC (Inter-Process Communication) on the phone via URL schemes.
 - Register a scheme (like `myapp://`), and then other apps can call it and your app will get launched, then you handle it.
 - Not true IPC, but close?
- Use the `CFBundleURLTypes` key in the Info.plist, then for each type, fill out `CFBundleURLName` and `CFBundleURLSchemes`.
- When application calls your URL scheme, you handle it with your application delegate: `application:handleOpenURL:`

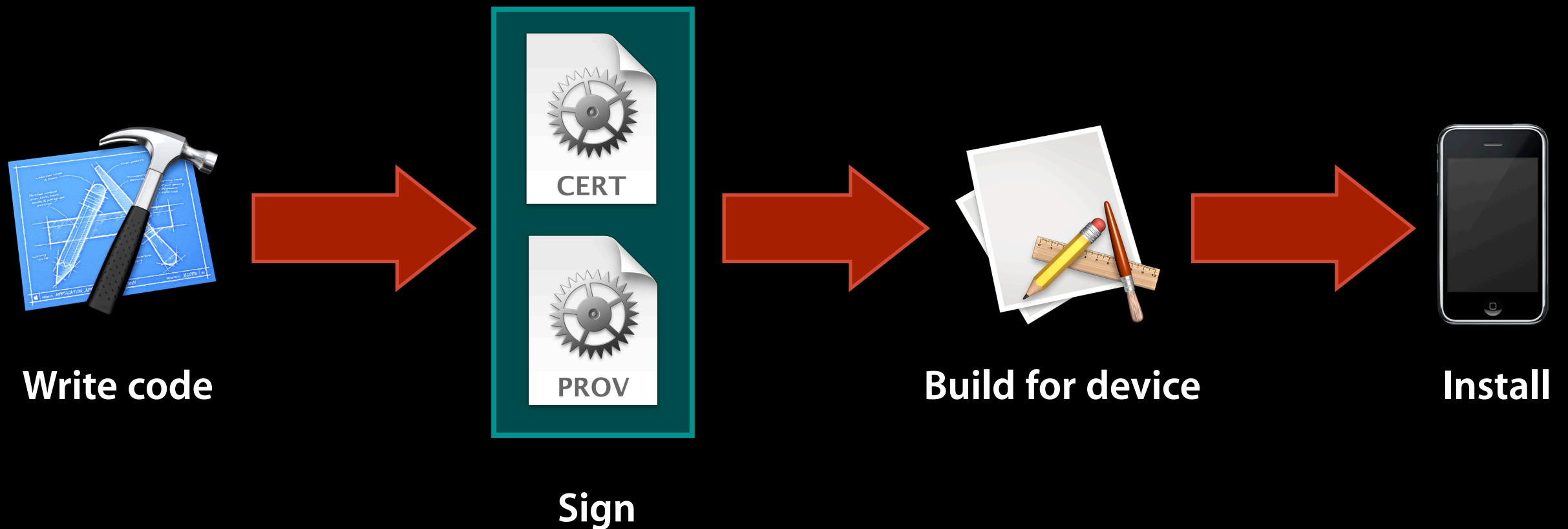
Introduction to iPhone Development

Contents		Task
1	Application Runtime Core Architecture and Life-cycles	
2	What's in a bundle? The resources in an app bundle	
3	Customizing Behavior How does it launch? IPC?	
4	Relation with Tools How do we get the app on the phone?	





Introduction to iPhone Development

Contents		Task
1	Application Runtime Core Architecture and Life-cycles	
2	What's in a bundle? The resources in an app bundle	
3	Customizing Behavior How does it launch? IPC?	
4	Relation with Tools How do we get the app on the phone?	





Development to Device



Introduction to iPhone Development

Contents		Task
1	Application Runtime Core Architecture and Life-cycles	
2	What's in a bundle? The resources in an app bundle	
3	Customizing Behavior How does it launch? IPC?	
4	Relation with Tools How do we get the app on the phone?	

Introduction to iPhone Development

Contents		Task
1	Application Runtime Core Architecture and Life-cycles	
2	What's in a bundle? The resources in an app bundle	
3	Customizing Behavior How does it launch? IPC?	
4	Relation with Tools How do we get the app on the phone?	

iPhone Application Programming Guide

Available on Apple's website

