

PhoneGap

PhoneGap is a free and open source framework that allows you to create mobile apps using standardized web APIs for native platforms (iPhone, Android, etc.)

Phonegap is a bridge between HTML/CSS/JS and native code.

- Code your app as regular web app
- Connect with PhoneGap API to add native system functionality

Knowledge of native code (Objective-C / Java) is not necessary but can be helpful

PhoneGap

Developed by Nitobi Software

Bought by Adobe

PhoneGap is the commercial name for Apache Cordova

PhoneGap/Cordova was contributed to Apache Software Foundation by Adobe

<https://cordova.apache.org/>

Free

+

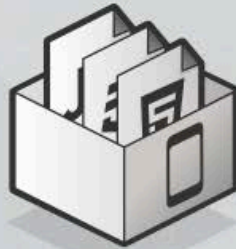
Based on open source standards

Native APIs

1. Build your app based on open web standards
2. Wrap with PhoneGap
3. Deploy to multiple platforms



Wrap your app with
PhoneGap



Deploy to **mobile
platforms!**



PhoneGap

is a framework for mobile app development using standard web technologies:

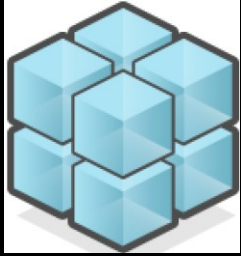
HTML/HTML5 – content

CSS/ CSS3 – style

JavaScript - accessing device functionality, logic,
interactivity

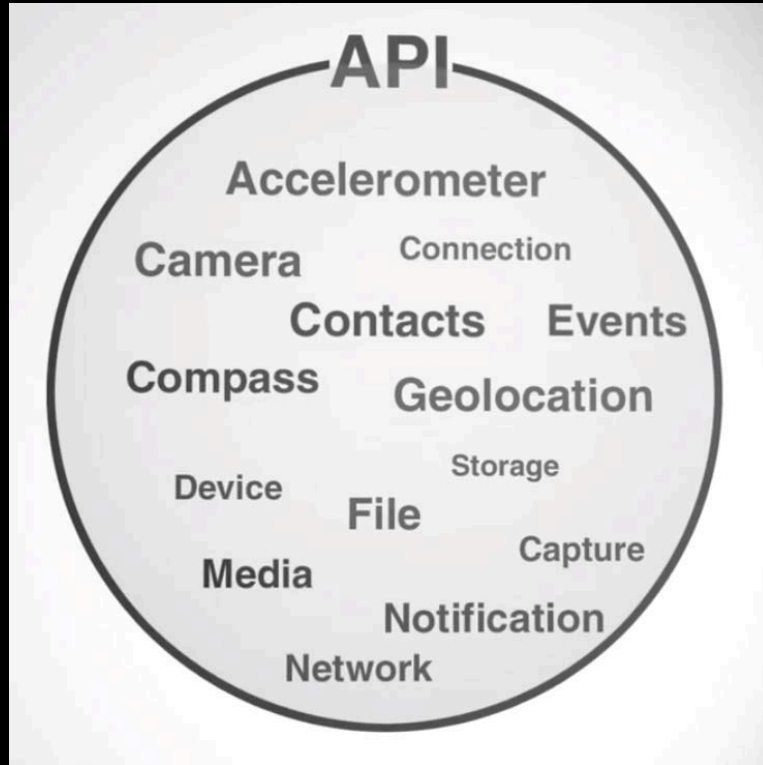


PhoneGap



Access native features

PhoneGap



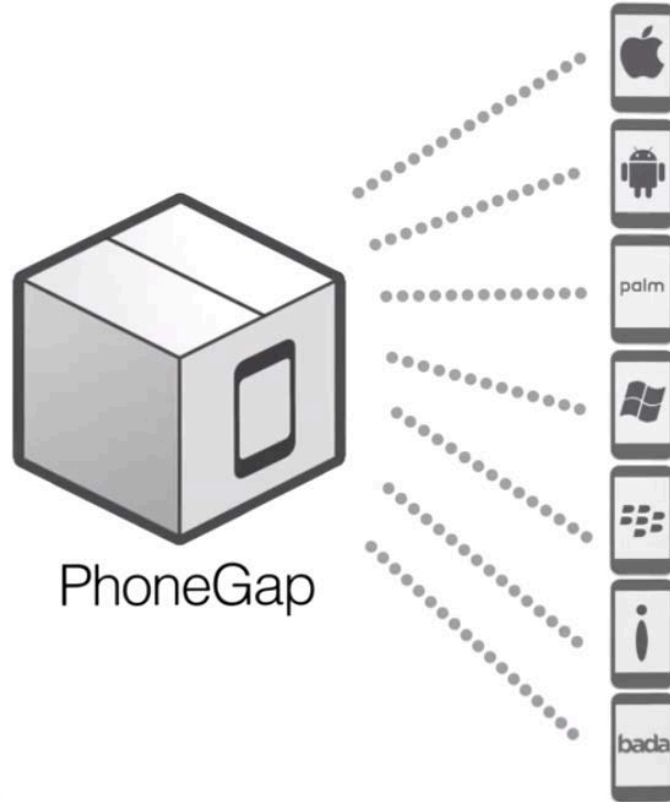
Native APIs

PhoneGap

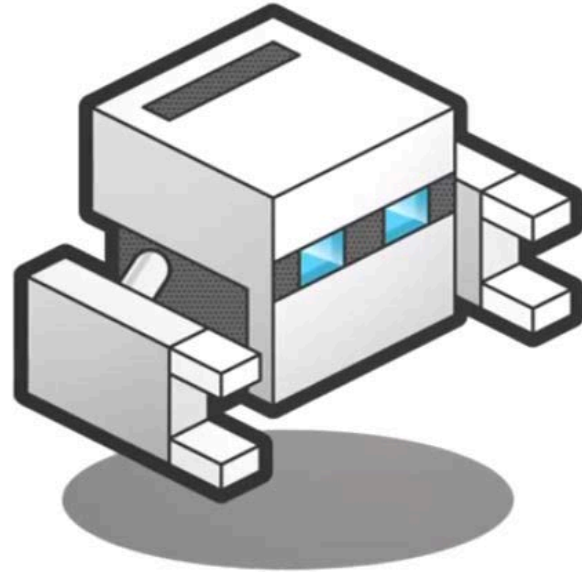


Deploy your app to Multiple Platforms

PhoneGap



PhoneGap



PhoneGap **:Build**

PhoneGap

The PhoneGap applications are hybrid

- They are neither truly native nor purely web based
 - All layout rendering is done via the web view instead of Objective-C
 - Much of the functions of HTML5 are supported

A disadvantage is that hybrid applications do not have full access to the device APIs

Camera, compass, accelerometer, etc.

PhoneGap

PhoneGap is a library that you must include in your app

- JavaScript and xml files

How PhoneGap works?

- PhoneGap generates a out-of-the-browser window that executes the HTML and JavaScript
- Due to a couple of xml and jar/dll files it enables the usage of native APIs

PhoneGap

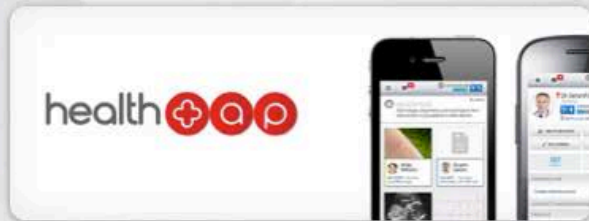
PhoneGap applications are web pages that access mobile device functionality (i.e. iPhone)

CSS transitions and animations are hardware accelerated on the iPhone, so Apps can provide a rich experience

HTML5 Canvas element used in games built with PhoneGap

PhoneGap – Example Apps

<http://phonegap.com/app/>



Native APIs

Geolocation

Contacts

Camera

Accelerometer: Access to device motion sensors

Device: Gather device specific information

Network: Quickly check the network state

Events: Hook into native events through JavaScript

Notification: Visual and audible device notifications

File: Hook into native file system through JavaScript

Native APIs

API Reference Documentation

<http://docs.phonegap.com/en/1.9.0/index.html>

Supported Features

	iPhone / iPhone 3G	iPhone 3GS and newer	Android	Blackberry OS 6.0+	Blackberry 10	WebOS	Windows Phone 7 + 8	Symbian	Bada
Accelerometer	✓	✓	✓	✓	✓	✓	✓	✓	✓
Camera	✓	✓	✓	✓	✓	✓	✓	✓	✓
Compass	X	✓	✓	X	✓	✓	✓	X	✓
Contacts	✓	✓	✓	✓	✓	X	✓	✓	✓
File	✓	✓	✓	✓	✓	X	✓	X	X
Geolocation	✓	✓	✓	✓	✓	✓	✓	✓	✓
Media	✓	✓	✓	X	✓	X	✓	X	X
Network	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Alert)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Sound)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Vibration)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Storage	✓	✓	✓	✓	✓	✓	✓	X	X

✓ - supported feature

X - unsupported feature due to hardware or software restrictions

Native APIs

PhoneGap provides a bridging library to access the device features and handles communication with the native OS

Many JS frameworks started to emerge for mobile devices: jQuery, JQM, Sencha Touch, Dojo, etc.

PhoneGap Custom Plugins

Extensible with a native plugin model that enables you to write your own native logic to access via JavaScript

Open source plugins at:

<https://github.com/phonegap/phonegap-plugins>

PhoneGap intro

Getting Started Guides

<http://docs.phonegap.com/en/3.4.0/index.html>

Phonegap google groups

<https://groups.google.com/forum/#!forum/phonegap>

PhoneGap example

```
function capturePhoto() {  
  navigator.camera.getPicture(onPhotoDataSuccess, onFail, {  
    quality: 50 });  
}  
  
function onPhotoDataSuccess(imageData) {  
  var smallImage = document.getElementById('smallImage');  
  smallImage.style.display = 'block';  
  smallImage.src = "data:image/jpeg;base64," + imageData;  
}  
  
function onFail(message) { alert('Failed because: ' + message); }
```

PhoneGap example

```
<body onload="onLoad()">
```

```
<button onclick="capturePhoto();">Capture Photo</button>  
<br>
```

```
<img style="display:none;width:60px;height:60px;"  
id="smallImage" src="" />
```

```
</body>
```

```
</html>
```

PhoneGap

You use the same web view of the native OS

iOS = `UIWebView`

Android = `android.webkit.WebView`

Why PhoneGap?

- Develop once, run in many devices
- Use familiar web tools (HTML, CSS, JavaScript)
- Reduced number of required skills
- Reduced development cost
- Reduced maintenance cost

Pros and Cons

HTML5, CSS3 and JavaScript skills

Cross platform

Single code base for all platforms

- iOS, Android, WP 7, etc.

Objective-C, Java and C# skills

Poor performance

- If the app is graphically intense, i.e. a game
- Caching or leveraging some 3rd party solutions may be implemented with native graphics
- For the most part a native app is much faster/smoothier than mobile web app

Pros and Cons

Takes advantage of distribution stores:

- Apple Store
- Android Market

Quicker way to the market in general than with native apps

Lack of pre-built UI transitions, standard controls, widgets

Therefore the development time can take longer

Ex. For a polished-looking app with a native look and feel

PhoneGap - Links

Phonegap Installation

http://docs.phonegap.com/en/3.0.0/guide_cli_index.md.html#The%20Command-line%20Interface

Phonegap Documentation

<http://docs.phonegap.com/en/3.4.0/index.html>

iOS Platform Guide

http://docs.phonegap.com/en/3.0.0/guide_platforms_ios_index.md.html#iOS%20Platform%20Guide

iOS Developer Center

<https://developer.apple.com/devcenter/ios/index.action>

PhoneGap - Export to Device

1. install SDKs for the native device platform (iOS SDK)
2. Otherwise, you can use the cloud-based PhoneGap Build service to compile apps

PhoneGap - Export to Device

Install the SDK

1. from the [App Store](#), available by searching for "Xcode" in the App Store application
2. from [Apple Developer Downloads](#), which requires registration as an Apple Developer

PhoneGap - Export to Device

Install the phonegap command-line tool (CLI):

1. Download and install [Node.js](http://nodejs.org/)

<http://nodejs.org/>

PhoneGap – node.js

[Download](#) | [Docs](#) | [Blog](#) | [Community](#) | [Modules](#) | [Resources](#) | [Jobs](#) | [About](#)



Node.js is a platform built on [Chrome's JavaScript runtime](#) for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Current Version: v0.10.26

[INSTALL](#)

[DOWNLOADS](#)

[API DOCS](#)



NODE.JS ON THE ROAD

PRODUCTION NODE HITS THE PAVEMENT
GET INSPIRED & INVOLVED [LEARN MORE](#)

Fork Node on GitHub

PhoneGap - Export to Device

2. Install the phonegap utility

Use terminal to type in the command:

```
$ sudo npm install -g phonegap
```


PhoneGap - Export to Device

3. Create the app

Use terminal to type in the command:

```
$ phonegap create hello com.example.hello HelloWorld
```

PhoneGap - Export to Device

4. Build the app

Use terminal to type in the command:

```
$ cd hello
```

```
$ phonegap build ios
```

```
[phonegap] detecting iOS SDK environment...
```

```
[phonegap] using the local environment
```

```
[phonegap] compiling iOS...
```

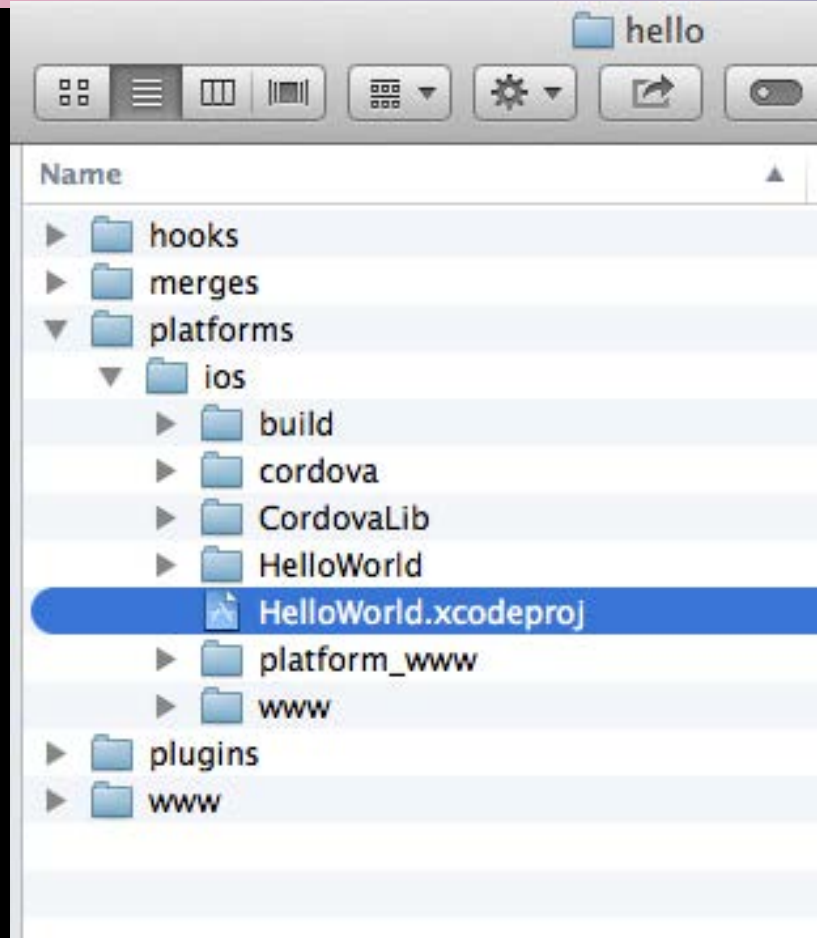
```
[phonegap] successfully compiled iOS app
```

PhoneGap - Export to Device

5. Test the app on device or simulator

```
$ phonegap install ios
```

PhoneGap - Export to Device

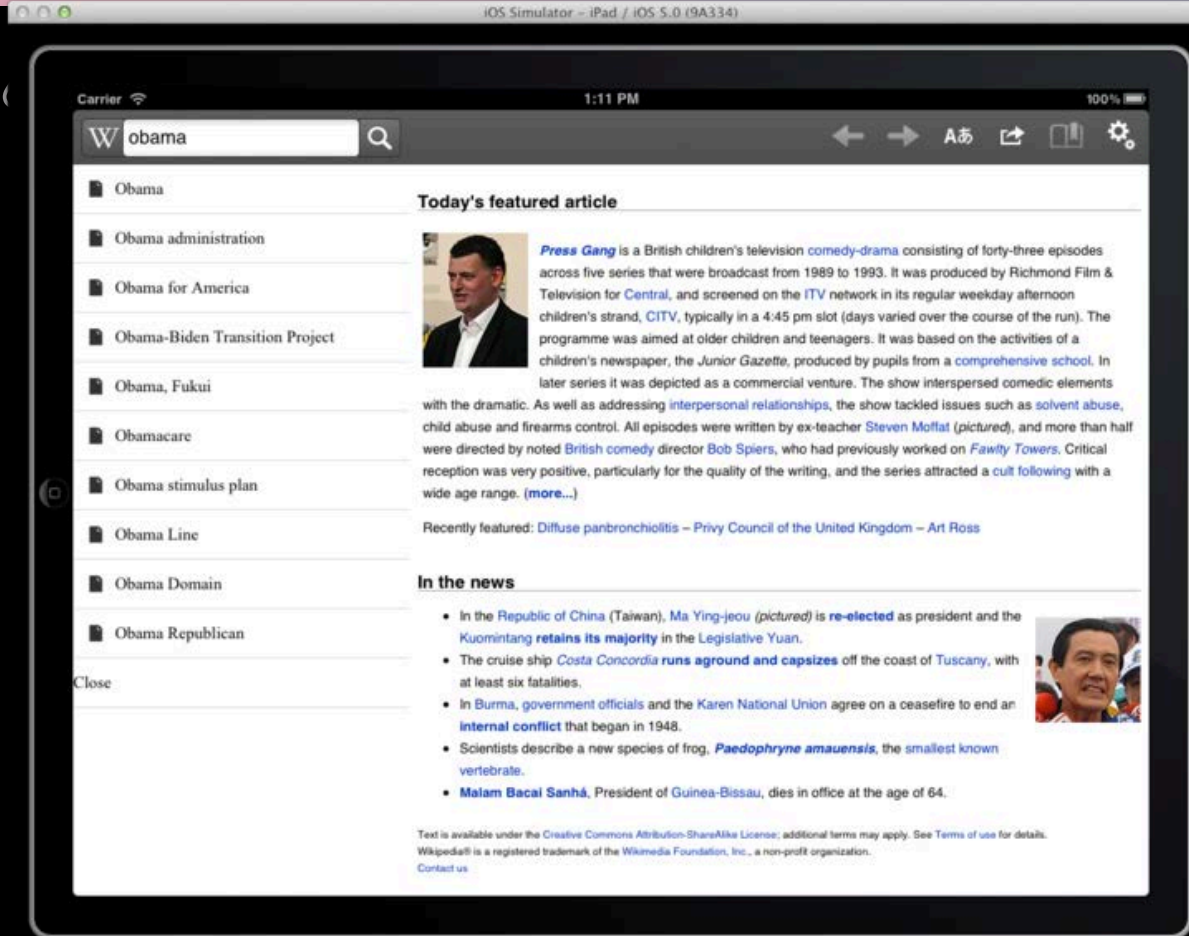


PhoneGap - Export to Device

6. Test the app on device or simulator

```
$ phonegap run ios
```

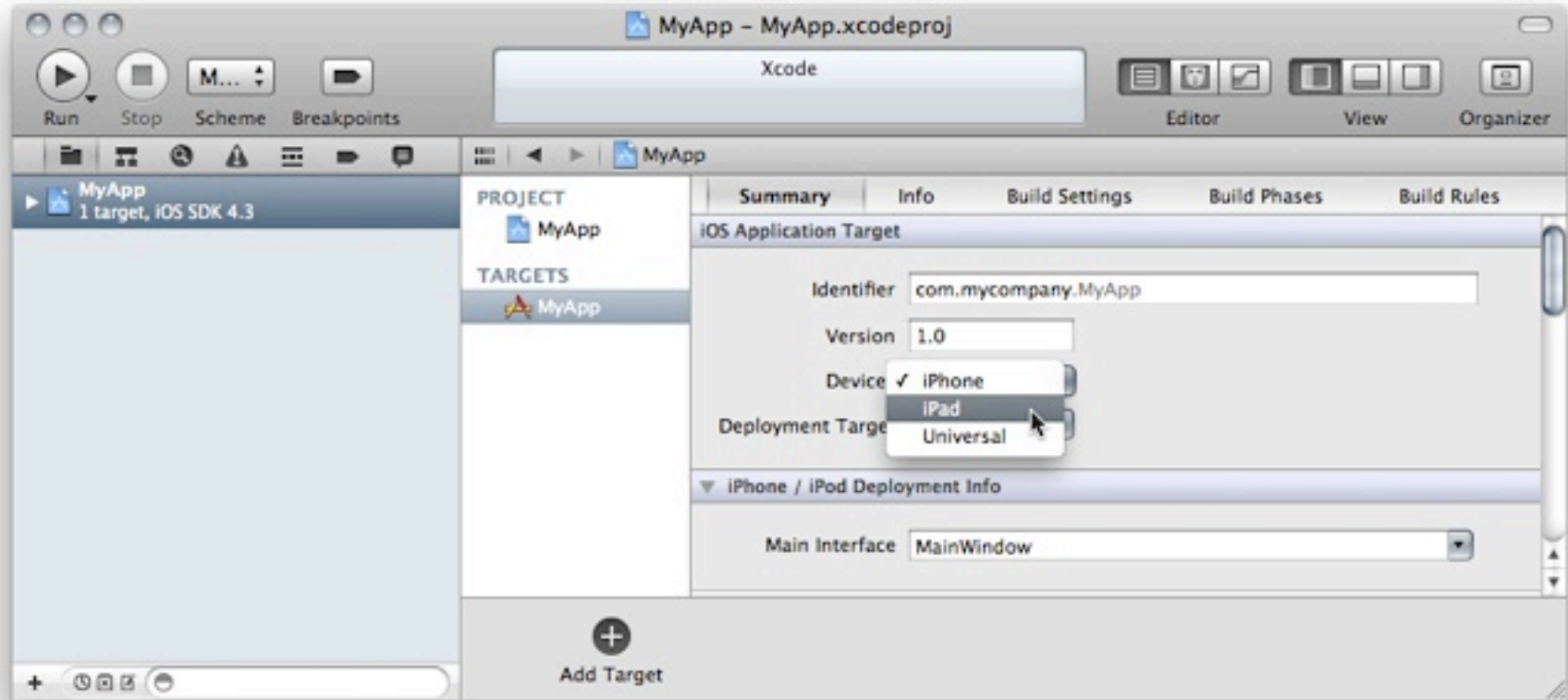
PhoneGap - Export to Device



PhoneGap - Export to Device

Open a Project in the SDK (Xcode)

PhoneGap - Export to Device



PhoneGap - Export to Device

Deploy to Simulator

- Make sure the `.xcodeproj` file is selected in the left panel.
- Select the hello app in the panel immediately to the right.
- Select the intended device from the toolbar's Scheme menu, such as the iPhone 6.0 Simulator
- Press the Run button. That builds, deploys and runs the application in the emulator.

PhoneGap - Export to Device

Deploy to connected device

- Join the Apple iOS Developer Program.
- Create a *Provisioning Profile* within the [iOS Provisioning Portal](#). You can use its *Development Provisioning Assistant* to create and install the profile and certificate Xcode requires.
- Verify that the *Code Signing* section's *Code Signing Identity* within the project settings is set to your provisioning profile name.

Project Navigator:

- HelloWorld
 - 1 target, iOS SDK 7.0
 - config.xml
 - www
 - merges
 - Staging
- CordovaLib.xcodeproj
 - 1 target, iOS SDK 7.0
 - Classes
 - Plugins
 - Other Sources
 - Resources
 - Frameworks
 - Products

Build Settings for CordovaLib

Build Settings | Build Phases | Build Rules

Basic | All | Combined | Levels

Any iOS Simulator SDK: i386

iphones7.* SDK: armv7 armv7s

iphones6.* SDK: armv7 armv7s

Build Options

Setting: CordovaLib

Compiler for C/C++/Objective-C: Default compiler (Apple LLVM 5.0)

Deployment

Setting: CordovaLib

Installation Build Products Location: /tmp/CordovaLib.dst

Installation Directory: /usr/local/lib

Skip Install: Yes

Strip Debug Symbols During Copy: <Multiple values>

 Debug: No

 Release: Yes

Strip Linked Product: Yes

Targeted Device Family: iPhone/iPad

iOS Deployment Target: iOS 6.0

Packaging

Setting: CordovaLib

Info.plist File: [empty]

Product Name: Cordova

Public Headers Folder Path: include/Cordova

Search Paths

Setting: CordovaLib

Always Search User Paths: No

Apple LLVM 5.0 - Code Generation

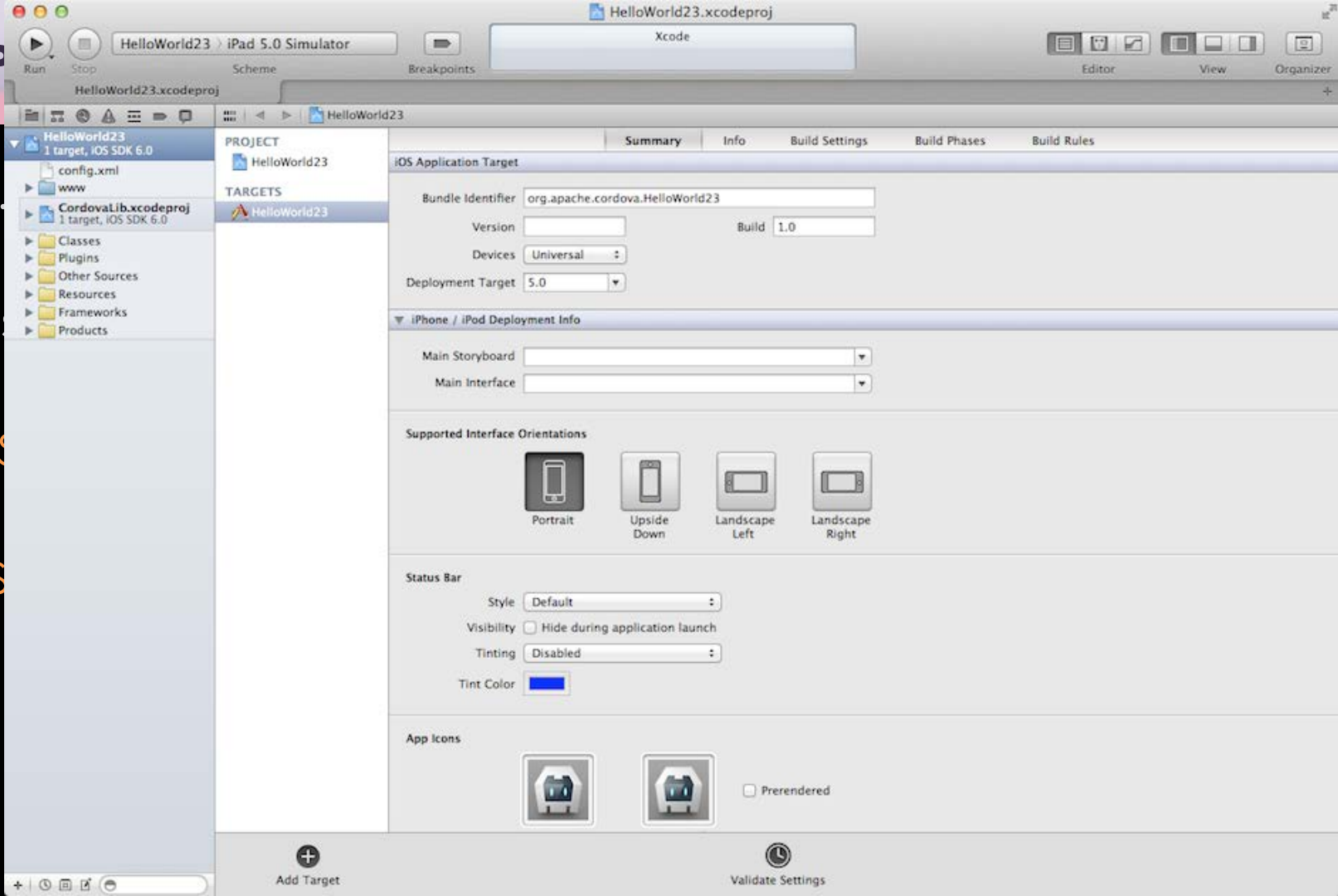
Quick Help

Declaration: GCC_VERSION

Description: The compiler to use for C, C++, and Objective-C.

Buttons:

- Push Button - Intercepts mouse-down events and sends an action message to a target object when it's...
- Gradient Button - Intercepts mouse-down events and sends an action message to a target object...



4

U

\$

lof

PhoneGap - Export to Device

Deploy to connected device

- Use the USB cable to plug the device into your Mac.
- Select the name of the project in the Xcode window's Scheme drop-down list.
- Select your device from the Device drop-down list.
- Press the Run button to build, deploy and run the application on your device.

PhoneGap - Export to Device

The screenshot shows the Xcode Organizer window with the following details:

- LIBRARY:** Device Logs, Screenshots
- DEVICES:** My Mac (10.9 (13A603)), dt iPhone (6.1.3 (10B329))
- dt iPhone Details:**
 - Capacity: 14.45 GB
 - Model: iPhone 5 (Model A1428)
 - Serial Number: F17JK6NSDTTP
 - ECID: 4164425035213
 - Identifier: 1d2f0ba55c93499b6f547baab7461f6a0b4f2ede
 - Software Version: 6.1.3 (10B329)
 - Restore using iTunes... (button)
 - Provisioning: iOS Team Provisioning Profile: * (dropdown)
 - Applications: HelloWorld (dropdown), 8 FairPlay-encrypted applications
 - Device Logs: (dropdown)
 - Screenshots: No screenshots (dropdown)

PhoneGap - Links

Phonegap Installation

http://docs.phonegap.com/en/3.0.0/guide_cli_index.md.html#The%20Command-line%20Interface

Phonegap Documentation

<http://docs.phonegap.com/en/3.4.0/index.html>

iOS Platform Guide

http://docs.phonegap.com/en/3.0.0/guide_platforms_ios_index.md.html#iOS%20Platform%20Guide

iOS Developer Center

<https://developer.apple.com/devcenter/ios/index.action>

PhoneGap App Icon

The iOS platform specifies:

72-pixel-square icons for iPads

57-pixel icons for iPhones and iPods

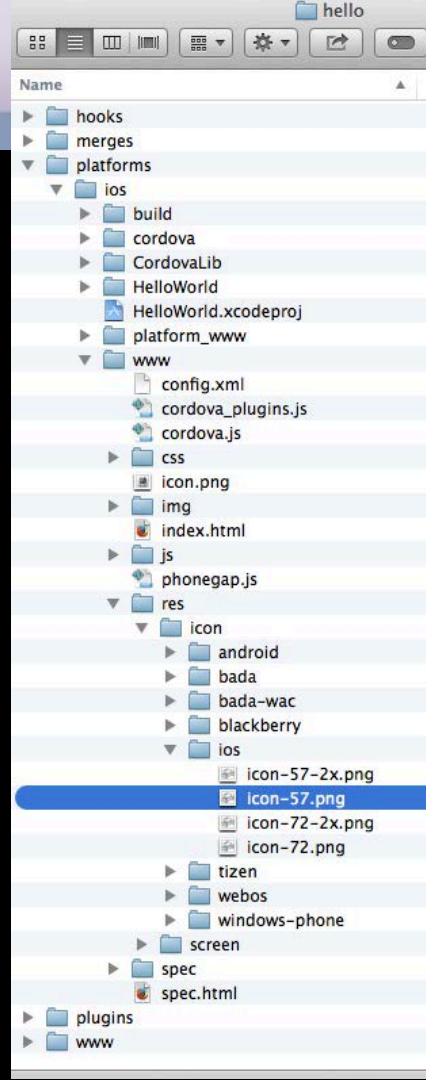
with high-resolution 2x variants for retina displays

`ios/icon-57-2x.png`

`ios/icon-57.png`

`ios/icon-72-2x.png`

`ios/icon-72.png`



PhoneGap Splash Screen

Use the Splashscreen API to enable display of an app's introductory splash screen.

In the CLI splash screen source files are in `www/res/screens` subdirectory

`ios/screen-ipad-landscape-2x.png`

`ios/screen-ipad-landscape.png`

`ios/screen-ipad-landscape-2x.png`

`ios/screen-ipad-landscape.png`

`ios/screen-iphone-landscape-2x.png`

`ios/screen-iphone-landscape.png`

`ios/screen-iphone-landscape-2x.png`

`ios/screen-iphone-landscape.png`

