# General information when developing for iOS

## Overview

This tutorial presents general development information for the iOS environment in the following topics:

- Project structure
- Designing for iOS
- Environment-specific images
- Accessing native capabilities by using Apache Cordova
- Optimizing applications
- Deployment

## Project structure

An iPhone/iPad application that is developed by using IBM MobileFirst Platform comprises the following components:

- An Objective-C ("native") Main App Delegate is provided and hosts an instance of a UIWebView component.
- `www.bundle` files, which are written in HTML, CSS and JavaScript, and which run in the UIWebView instance.
- A set of Objective-C and JavaScript libraries that provide access to various iOS features and capabilities.
- All application components, including the web code that the developer provides, are packaged into a single universal iOS application.

If you use a source control management system (such as Rational Team Concert, Git, Subversion and so on...), see the topic about integrating with source control system, in the user documentation.

## Designing for iOS

### Guidelines

Starting with iOS7, the design language of the operating system has changed drastically.
It is therefore highly encouraged and useful for developers to consult the Apple iOS Human Interface Guidelines on the Designing for iOS (https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/) page.

When developing for iOS, it is also useful to consult the iOS Developer Library (https://developer.apple.com/library/ios/navigation/ ) page.

### Resolutions

iOS devices have various screen resolutions:

- iPhone 1/2/3/3GS – 320 x 480
- iPhone 4/4S – 640 x 960
- iPhone 5/5s – 640 x 1136
- iPhone 6 - 1334 x 750
- iPhone 6 plus - 1920 x 1080
- iPad 1/2/mini – 768 x 1024
- iPad 3/4/5 – 1,536 x 2,048

When displaying graphical elements on wider screens, iOS automatically scales images and fonts to the appropriate ratio.
For example, on a 640-pixel wide screen, a 100-pixel wide image is scaled by 640/320 (= 2) to 200 pixels in width, and the text that is displayed with an 18-pixel font is displayed with a 36-pixel font.

### Developer's actions

As a developer, you are therefore responsible for the following adjustments:

- Ensure that images look good at all resolutions:
  - Provide images for the environment with the highest resolution requirements.
  - In the CSS, scale the image width to the lower resolution.
    In the previous example, provide an image with a width of 200 pixels, and in the CSS, scale it to 100 pixels. iOS scales it back by a factor of 2 to be 200 pixels wide.
- Make sure that text is displayed appropriately at all resolutions.
- Consider specifying font sizes in ems, rather than in pixels.

## Environment-specific images

Image file sizes differ depending on your iDevice. For more information, see the Apple iOS UI Development Guidelines for icon and image sizes (https://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/IconMatrix.html#//apple_ref/doc/uid/TP40006556-CH27-SW1) page.

## Accessing native capabilities by using Apache Cordova

You can use the Apache Cordova framework in iOS applications to access the native elements of a device, such as contacts, geolocation services, media services, or the accelerometer.

For more information about Apache Cordova development, see the Apache Cordova overview (../../adding-native-functionality/apache-cordova-overview) tutorial.

## Optimizing applications

When developing a mobile application, you can use minification and concatenation to reduce the size and number of files that are used within the application. This feature is available for the following environments: Android, iOS, Windows 8, Windows Phone 8, BlackBerry 10, Mobile Web, and Desktop Browser.

> For more information about minification and concatenation, see the topic about optimizing MobileFirst applications, in the user documentation.

## Deployment

iOS application deployment is done through the Apple App Store.

**Important:** Before you build a production version of your application, make sure that the `worklightSettings` property is set to `false` in the `application-descriptor.xml` file.

1. Follow the App Store approval process.
2. Manage your apps on the App Store.