Document name	**iOS Web Applications for native look and feel**
Version no.	1.0
Release date	15-Mar-2012

iOS Web Applications for native look and feel

Table of Contents

1. Introduction

This will be a reference document for the Configuring iOS Web Applications for native look and feel and iPhone 4 Retina Display.

2. Configuring iOS Web Applications for native look and feel

A web application is designed to look and behave in a way similar to a native application—for example; it is scaled to fit the entire screen on iOS. You can tailor your web application for Safari on iOS even further, by making it appear like a native application when the user adds it to the Home screen. You do this by using settings for iOS that are ignored by other platforms.

2.1 Specifying a Webpage Icon for Web Clip

- You may want users to be able to add your web application or webpage link to the Home screen. These links, represented by an icon, are called Web Clips. Follow these simple steps to specify an icon to represent your web application or webpage on iOS.
- To specify an icon for the entire website (every page on the website), place an icon file in PNG format in the root document folder called apple-touch-icon.png or apple-touch-icon-precomposed.png. If you use apple-touch-icon-precomposed.png as the filename, Safari on iOS won't add any effects to the icon.
- To specify an icon for a single webpage or replace the website icon with a webpage-specific icon, add a link element to the webpage, as in:

```
<link rel="apple-touch-icon" href="/custom_icon.png"/>
```

- In the above example, replace custom_icon.png with your icon filename. If you don't want Safari on iOS to add any effects to the icon, replace apple-touch-icon with apple-touch-icon-precomposed.
- To specify multiple icons for different device resolutions—for example, support both iPhone and iPad devices—add a sizes attribute to each link element as follows:

```
<link rel="apple-touch-icon" href="touch-icon-iphone.png" />
<link rel="apple-touch-icon" sizes="72x72" href="touch-icon-ipad.png"
/>
<link rel="apple-touch-icon" sizes="114x114" href="touch-icon-
iphone4.png" />
```

- The icon that is the most appropriate size for the device is used. If no sizes attribute is set, the element's size defaults to 57 x 57.
- If there is no icon that matches the recommended size for the device, the smallest icon larger than the recommended size is used. If there are no icons larger than the recommended size, the largest icon is used. If multiple icons are suitable, the icon that has the precomposed keyword is used.
- If no icons are specified using a link element, the website root directory is searched for icons with the apple-touch-icon... or apple-touch-icon-precomposed... prefix. For example, if the appropriate

icon size for the device is 57 x 57, the system searches for filenames in the following order:

1) apple-touch-icon-57x57-precomposed.png
2) apple-touch-icon-57x57.png
3) apple-touch-icon-precomposed.png
4) apple-touch-icon.png

***iOS Note: The Web Clip feature is available in iOS 1.1.3 and later. The apple-touch-icon-precomposed.png filename is available in iOS 2.0 and later. Support for multiple icons for different device resolutions is available in iOS 4.2 and later.**

2.2 Specifying a Startup Image

- On iOS, similar to native applications, you can specify a startup image that is displayed while your web application launches. This is especially useful when your web application is offline. By default, a screenshot of the web application the last time it was launched is used. To set another startup image, add a link element to the webpage, as in:

```
<link rel="apple-touch-startup-image" href="/startup.png">
```

- In the above example, replace startup.png with your startup screen filename. On iPhone and iPod touch, the image must be 320 x 460 pixels and in portrait orientation.

IOS Note: **Specifying a startup image is available in iOS 3.0 and later.**

2.3 Hiding Safari User Interface Components

As part of optimizing your web application for Safari on iOS, have it launch in full-screen mode to look like a native application. When using full-screen mode, Safari is not used to display the web content—specifically, there is no browser URL text field at the top of the screen or button bar at the bottom of the screen. Only a status bar appears at the top of the screen. Read "**Changing the Status Bar Appearance**" for how to minimize the status bar.

Set the apple-mobile-web-app-capable Meta tag to yes to turn on this feature. For example, the following HTML displays web content in full-screen mode.

```
<meta name="apple-mobile-web-app-capable" content="yes" />
```

You can determine whether a webpage is displayed in full-screen mode using the window.navigator.standalone read-only Boolean JavaScript property.

2.4 Changing the Status bar Appearance

If your web application displays in full-screen mode like that of a native application, you can minimize the status bar that is displayed at the top of the screen on iOS. Do so using the status-bar-style meta tag.

This meta tag has no effect unless you first specify full-screen mode as described in “Hiding Safari User Interface Components.” Then use the status bar style meta tag, apple-mobile-web-app-status-bar-style, to change the appearance of the status bar depending on your application needs. For example, if you want to use the entire screen, set the status bar style to translucent black.

For example, the following HTML sets the background color of the status bar to black:

```
<meta name="apple-mobile-web-app-status-bar-style" content="black" />
```

Reference:
https://developer.apple.com/library/ios/#DOCUMENTATION/AppleApplications/Reference/SafariWebContent/ConfiguringWebApplications/ConfiguringWebApplications.html

3. iPhone 4 Retina Display

The iPhone 4’s eye-popping high-resolution screen.
For three years, the iPhone came in just one flavor of screen resolution: the 3.5-inch screen was always 320x480 pixels. The new phone doubles the resolution to 640x960 pixels.

Figure 1 Retina Display

3.1 Pixels to Points

Starting in iOS 4, dimensions are measured in “points” instead of pixels. Conveniently enough, the iPhone screen is 320x480 points on both iPhone 4 and older models. Since that matches the pixel dimensions on older phones, existing apps look and work the same on those phones in iOS 4 as they did in earlier operating systems. There, one pixel is one point, a one-to-one match. In other words, dimensions for all the various elements of iOS 4 remain the same, but their units have changed: you just substitute points where you used to say pixels.

On iPhone 4, a point is two pixels; draw a one-point line, and it shows up two pixels wide. **So just specify your measurements in points for all devices, and iOS automatically draws everything to the right proportion on the screen.** Text and images remain the same physical size on both old and new phones. That goes for bitmap images in legacy apps, too**; iOS 4 blows them up, automatically pixel-doubling them to adapt to the new phone’s resolution.**

3.2 Handling Images in Native Applications

Of course, pixel-doubled images don't take advantage of the gloriously crisp display on the new phone. That's where your extra legwork comes in: to add high-resolution images to your app, you have to include a second set of all your graphic files. For every image in your app, add a second version that's twice the size, adding @2x to the name. For a low-resolution image named image.png, for example, you would add a second file named image@2x.png. The new image will be picked up automatically by iPhone 4. Everywhere your code requests image.png (or even just plain old image), image@2x.png will be used instead.

iOS 4 goes seamlessly in the other direction (scaling up low-res images for iPhone 4), it doesn't work the other way. By default, it scales all images so that one pixel equals one point. That's a great solution for making sure that old apps work correctly on the new phone, but it doesn't help you go the other way 'round.

3.3 Handling Images in Web Applications

3.3.1 iPhone 4 Retina Detection

This is a big issue for statistics and maybe if you are thinking of a special high-DPI version for iPhone 4.

JavaScript objects return same values and the User Agent.

This is the User Agent for iPhone 4 and classic iPhone/iPhone 3GS with iOS 4:

Mozilla/5.0 (iPhone; U; CPU iPhone OS 4_0 like Mac OS X; en-us) AppleWebKit/532.9 (KHTML, like Gecko) Version/4.05 Mobile/8A293 Safari/6531.22.7

CSS Media queries can be used to specify a separate css file for the iPhone 4 retina display

Method-I

Using the CSS Media Query Specify a separate css file for the retina device

```
<link rel="stylesheet" href="highres.css" media="only screen and (-webkit-
min-device-pixel-ratio: 2), only screen and (min-device-pixel-ratio: 2)" />
```

Method-II

Enclose Retina specific css in a separate code block in the same css file

```
@media all and (-webkit-min-device-pixel-ratio:2) {
    .bg {
        background-image: url(chevron-hd.png);
    }
}
```

Method-III

The same can be achieved using java script

```
if (window.devicePixelRatio == 2)
 {
                        //Targeted code goes here
}
```

To scale down background images use the new -webkit-background-size CSS For instance, using an proper image which is 32x16 pixels, you'll have to produce a 64x32 new image and use:

```
.bg {

    background-image: url(chevron.png);
    -webkit-background-size: 32pt 16pt;
}
```