Porting iPhone web applications to S60 5th Edition

Introduction

This article is meant to provide beginning and intermediate iPhone WebApps Developers with an introduction to porting applications to S60 5th edition, for both mobile browser and WRT Widget platform.

General Considerations

iPhone (and iPod Touch) browser is Mobile Safari, based on the open source WebKit browser. S60 5th edition devices (as 5800 Xpress or N97) uses the same base (WebKit) for both the mobile browser and the widget engine (WRT), providing a similar experience in XHTML and CSS rendering using standards.

Solutions

In an iPhone with mobile technologies you can, besides a normal HTML website, create:

- A mobile optimised website for iPhone and other mobile devices
- A web application for iPhone. An application compatible only with iPhone using extensions.

For all the possibly solutions you can add an icon in the Home Screen to access the web and a web application.

There is also a hybrid solution that involves a native iPhone application to embed a web application to be distributed in AppStore using a UIWebView class or using the PhoneGap open source solution. PhoneGap supports iPhone, Android, Blackberry, S60 (through WRT emulation layer) and Maemo.

In a S60 5th Edition with mobile technologies you can, besides a normal HTML website, create:

- A mobile optimised website for S60 and other mobile devices
- A web application optimized for S60 5th edition only
- A Web Runtime (WRT) widget, an installed web application in the Applications menu.

Compatibility

Compatible Technologies in both platforms

- XHTML 1.0/1.1 and XHTML MP (Mobile Profile)
- CSS 2.1
- JavaScript 1.5
- XMLHttpRequest object for AJAX
- URL Schemes for SMS and mail sending

iPhone Webapps non-compatible features

iPhone mobile browser has several additional features to the standards not compatible with S60 5th edition or other mobile device. Here is the list and we'll cover how to port each feature to S60 5th edition later.

- Client-side storage with HTML 5 features
- Canvas for vector drawing
- URL Schemes for communicating with other applications
- CSS Visual Effects
- Icon for Home Page
- Viewport metadata
- Touch events
- CSS 3.0 Features

S60 5th Edition web browser non-compatible features

The S60 browser has some additional features not compatible with iPhone webapps.

- Flash support
- iFrame support
- URL Schemes for communicating with other applications

(C) Copyright Nokia 2013. All rights reserved.

S60 5th Edition WRT widgets non-compatible features

The WRT widget platform has some additional features not compatible with iPhone webapps.

- Flash support
- iFrame support
- URL Schemes for communicating with other applications
- Localized name and icon for Applications Menu
- Navigation features for hide or show a cursor, allows tab navigation and create transitions between divs.
- JavaScript Service API
- JavaScript Widget API
- Home screen support (only for some compatible devices, as Nokia N97)

Porting features for both S60 browser and WRT widget

For create a S60 browser web application or a WRT widget for S60 5th edition from an iPhone webapp, you need to migrate the following code.

Canvas for vector drawing

S60 browser (and WRT) doesn't support the <canvas> tag in (X)HTML so we need to change that code. The canvas tag allows an iPhone web app to create, using JavaScript a low-level drawing zone in the webpage. The best way to port canvas drawing is to use Adobe Flash Lite technology.

With Adobe Flash Lite technology you can support every feature in canvas tag and a lot more. You can use Adobe Flash Professional to create all the vector and bitmap drawing that you need in your website, fully compatible with S60 Browser and WRT Widgets.

Using ActionScript 2.0 or the timeline in design mode you can port the canvas javascript code.

Another solution is to create the drawing using the standard SVG-Tiny

---

URL Schemes for communicating with other applications

In an iPhone webapp, you can create URLs in links, using some URL schemes. The following list is the translation features between iPhone browser and S60/WRT browser:

Mail Scheme

Your e-mail scheme will work in S60 and WRT Browser. You don't need to change them. It will open the Mail application in the S60 device.

```
<a href="mailto:youruser@yourdomain.com">Contact us</a>
```

Phone Scheme

Your phone call scheme will work in S60 and WRT Browser as an alias for the WTAI scheme in S60 Browser. So, you don't need to change them. The device will call the phone number if the user accepts the alert window.

```
<a href="tel:number">Call us</a>
```

SMS Scheme

Your SMS text scheme will work in S60 and WRT Browser and you can add the text of the SMS. The device will open the SMS application with the destination number and the optional body typed.

```
<a href="sms:number">Text us</a>
```

You can add the body to your iPhone web app SMS link. Remember to generate a URL valid link using encodings, like %20 for any space in the link.

```
<a href="sms:number?body=This%20is%20the%20text">Text us</a>
```

Maps Scheme

In iPhone, you can link to a Google Maps website and the device will open the Maps Application. In S60 Browser, that link will still work opening the Google Maps mobile website instead of an application. So, you don't need to change your code.

YouTube Scheme

(C) Copyright Nokia 2013. All rights reserved.

In iPhone, you can link to a YouTube video and the device will open the YouTube application. In S60 Browser, that link will still work opening the YouTube mobile website instead of an application. So, you don't need to change your code. The S60 browser supports FLV and can view YouTube website without other application or closing the browser.

Font Support

S60 Web Browser and WRT font support applies to color and size. In an iPhone web application you can choose many font faces, that will not take effect in S60. The website will still work without any errors. You need to test how the different font face will render in your design.

CSS Visual Effects

The following CSS iPhone properties will not work in S60 browser. If you leave them, they won't affect your rendering

- -webkit-border-image
- -webkit-border-radius
- -webkit-text-size-adjust
- -webkit-tap-highlight-color

Touch Events

The following JavaScript events will not work in S60 browser nor WRT widget.

- ontouchstart
- ontouchend
- ontouchmove
- ontouchend
- ongesturestart
- ongestureend
- ongesturechange

You should recreate these events using the standard onclick, onmousedown and onmouseup. As the S60 devices are not multi-touch, the zoom and rotate gestures in the iPhone must be implemented in other ways, as a vertical slider for zoom and an icon for rotation.

Icon for Home Page

The home screen icon for iPhone is a Meta Tag that won't make any difference in the S60 website or WRT widget. You can leave or not the following tag if you are using it in your iPhone application:

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

You can replace, or add the following line to use the same icon as the favicon used in S60 browser

```
<link rel="shortcut icon" href="/path_to_icon.png">
```

Both Meta tags can live in the same HTML page. The iPhone will use the first one and the S60 browser will use the second. The PNG file must be a 57x57 pixeles to be compatible for iPhone. The S60 browser can use it or you can create a 16x16 ICO file and reduce the size of the image.

Viewport metadata

iPhone webapps supports the Viewport metadata to give information about the zoom and viewport that we want to be used in our website. S60 automatically adjust the viewport and there is no need and no support for this feature. If you create a WRT widget for S60 you can have a fullscreen web application as the fullscreen mode in an iPhone webapp.

Porting from iPhone webapp to S60 browser

Apart from Porting features for both S60 browser and WRT widget information, you can also consider the following porting situations for a S60 browser application.

Client-side storage with HTML 5 features

There isn't any client-side storage feature in S60 5th edition browser for your web applications. You can use a server-side storage solution, create a WRT widget for your web application or use cookies as your information package.

Icon for Home Page

If you want to have an icon for your website in the Applications Menu or Home screen in your S60 5th edition, you can use one of the following situations:

(C) Copyright Nokia 2013. All rights reserved.

- Create a WRT widget that opens a website. You must provide a link in your website to the widget, host the wgz file and it will automatically be installed in the user device Application menu.
- Add the website to Favorites and you can use it as a Home Screen Shortcut in Preferences of your phone.

Porting from iPhone webapp to WRT widget

Apart from Porting features for both S60 browser and WRT widget information, you can also consider the following porting situations for WRT widget application for S60 5th edition.

External links

For external links (to another website, to a YouTube video or to a Google Maps application) a webapp for iPhone uses the A tag in HTML. In WRT widget is better to open a new browser instance to the user access an external content outsite of your widget. To do it, you can use the widget.openURL method that leaves your widget on the background while the user browse the external content.

You can change every link:

```
<a href="http://www.cnn.com">Go To CNN</a>
```

With this code

```
<a href="javascript:widget.openURL('http://www.cnn.com')">Go To CNN</a>
```

Or, if you want to be unintrusive, you can leave the original link and cancel it if it's a WRT engine:

```
<a href="http://www.cnn.com" onclick="if (widget!=undefined) {widget.openURL(this.href); return false}">Go To CNN</a>
```

Client-side storage with HTML 5 features

In an iPhone webapp you can use HTML 5 features to generate SQL-like databases on the client. With WRT widgets you can use the Storing Settings API to store information on the device for future use from the widget.

```
// To store a variable
widget.setPreferenceForKey('value', 'name');

// To retrieve a variable
var data = widget.preferenceForKey('name');
```

The value has to be a string, so if in your iPhone webapp you were using tables with records you can recreate the same scheme using an array of objects in JavaScript and then converting them to a JSON-like string. The string can be converted again to an array of objects using eval method. If you are using Prototype or jQuery the library can help you converting the array to a JSON string. This is a sample:

```
// Class Friend definition
var Friend = function(n, i, e) {
    this.name = n;
    this.icon = i;
    this.email = e;
}

function quote(string) {
    return "'" + string.replace("'", "\'") + "'";
}

// Function to convert a Friend to JSON friendly
function friendToJSON(friend) {
     return "{ name: " + quote(friend.name) + ", icon: " +
                quote(friend.icon) + ", email: " +
                quote(email) +
            "}";
}


// Function to convert an array to JSON friendly
function arrayToJSON(array) {
    var json = "[";
    for (var i=0; i<array.length; i++) {
        json += array[i].toJSON() + (i==array.length-1) ? "" : ",";
    }
    json += "]";
    return json;
}

// Save friends "table" to the storage
function saveFriends() {
    widget.setPreferenceForKey(arrayToJSON(friends), 'friends');
}

// Load friends from the storage or create the storage for the first time
function loadFriends() {
    if (widget.preferenceForKey('friends')!=null) {
       // The storage already has a collection
       friends = eval("(" + widget.preferenceForKey('friends') + ")");
    } else {
       // Initialization for the first time
       friends = [
            new Friend('John', 'john.gif', 'john@developer.com'),
            new Friend('Mary', 'mary.gif', 'mary@developer.com')
```


(C) Copyright Nokia 2013. All rights reserved.

```
        ];
    }
}

// Similar to a "table" of friends
var friends;
```

Icon for Home Page

You can use the same icon as the iPhone Home Page icon if you are using or you can create another one for the S60 device to use in the Applications menu after the WRT widget is installed.

If in your iPhone webapp you have

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

You can use the same icon for your WRT widget renaming it to Icon.png in the root folder of you WRT widget. Remember you can localize that icon to many languages.

Viewport metadata

iPhone Viewport metadata is useless in a S60 5th edition WRT widget. You can leave or remove them from the source code. A widget is full-screen, so you don't need to specify that and it's better to have a widget designed to be contained in the screen without double scrollbar.

IDEs

If you created your iPhone web application using Aptana and the iPhone plugin it will be very easy to port to a WRT widget using the Nokia Web Tools (which are based on Aptana)

If you created your iPhone web application using Dashcode for Mac you can easily import the project into the Nokia Web Tools. Dashcode uses its own JavaScript component library optimized for the iPhone browser. It's difficult to create a porting modifying that library, so you can use WRTKit from Nokia to create a similar application cycle without HTML and CSS.

Additionals

Remember that you can add to your application, once is ported to a WRT widget the following features using Platform Services:

- Access to calendar
- Access to messaging
- Access to sensors
- Access to location
- Access to multimedia
- Access to Home Screen (some devices only)

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.