# How to port Bondi Widgets to Symbian Web Runtime

This article needs to be updated: If you found this article useful, please fix the problems below then delete the {{ArticleNeedsUpdate}} template from the article to remove this warning.

**Reasons:** hamishwillee (13 Oct 2011)
The Bondi site no longer exists - links to it have broken. It appears that Bondi may have been merged into WAC SDK however I can't find a clear explanation. This article therefore out of date and should either be deleted or updated to reflect the new API and state of Bondi. This will require some expert help.

## Preface

There are many different widget technologies and runtimes out there. One common denominator between many of them is the fact that they are, in effect, renamed zip packages containing common web technologies like HTML, CSS and Javascript etc.

Bondi is an open source industry collaboration for widget and web technologies. The following contains guidelines for porting a Bondi widget to the S60 Web Runtime.

## Target Audience

These instructions are targeted for developers who wish to port their own Bondi widgets to Nokia WRT. It is assumed that they know whether a Bondi widget uses built in Bondi objects and if it requires network access.

## Technology Compatibility

### Web standard technologies

Nokia WRT supports the following technologies:

- HTML 4.01 Specification, W3C Recommendation 24 December 1999,
- XHTML Mobile Profiles Specification, XHTML Mobile Profile Version 29-Oct-2001, WAP-277-XHTMLMP-20011029-a
- CSS Level 2 revision 1 (CSS 2.1), WAP CSS Specification Version 26-Oct-2001, WAP-239-WCSS-20011026-a
- JavaScript 1.5 (ECMA-262 3rd Edition), ECMAScript Language Specification, 3rd edition (December 1999)
- The combination of XML, XHTML/HTML, CSS, DOM, and the XMLHttpRequest object to add Ajax capability to a widget

If the widget you wish to port uses technologies that are not in this list, then it will not function as intended without modifications to the source code.

### Runtime specific Javascript APIs

If only standard Javascript has been used in the widget, then no additional javascript libraries need to be added to the widget.

However, the Bondi Widget environment provides Javascript APIs that allows widgets to access special functionality. Those functionalities work differently in the Nokia WRT environment. If the built-in Bondi Widget APIs have been used, then additional Javascript libraries need to be added to the widget. Step 4 explains how to do this.

Be aware that the library is only partially supported and as such will have reduced functionality. The following is a list of the supported functionality.

### Supported attributes and methods

- bondi.devicestatus.getPropertyValue({aspect:Battery,property:batteryBeingCharged})
- bondi.geolocation.getCurrentPosition()

To add more functionality, refer to Extending Bondi wrapper library.

## The Method

### 1. Unzip the Bondi package

(C) Copyright Nokia 2014. All rights reserved.

Take a Bondi widget and unzip it using the unzipping program of your choice. Remember to retain the folder structure, if possible.
Note: file extension may need to be changed to *.zip before being extracted.

## 2. Create an `info.plist` manifest file

When installing a *.wgz file the system looks for a manifest file called info.plist. The Bondi Widget equivalent is called `config.xml`.

Here's how to use the Bondi Widget `config.xml` data to create the WRT manifest file.

1. **Download and unzip this info.plist template file and save it in the widget's root directory.**

2. **Add the `DisplayName` (mandatory)**
   This can be found from the `<name>` element of the config.xml file.

3. **Add the `Identifier` (mandatory)**
   The `id` attribute in `config.xml` can be found in the `<widget>` element.

```
<widget xmlns="http://www.w3.org/ns/widgets/"
 xmlns:bondi="http://bondi.omtp.org/default.aspx/widgets"
 id="http://bondi.omtp.org/default.aspx/widgets/location"
 version="1.0">
```

These can be formatted into the same form as the conventional WRT Identifier, which is the URL in reverse with the widget name appended to the end. e.g. http://bondi.omtp.org/default.aspx/widgets/location → org.omtp.bondi.widgets.location

1. **Add the `MainHTML` (mandatory)**
   Bondi widget config.xml files contain a `<content>` element, whose attribute `src` contains the HTML file to be opened by the HTML renderer. If this element exists, its value should be assigned to the MainHTML key in the info.plist file.

   If it does not exist, then the default value, index.html should be assigned to the MainHTML key.

2. **`AllowNetworkAccess` (optional)**
   If the widget requires network access, add a self closing tag `<true />` under `AllowNetworkAccess` element. To deny network access, add a self closing tag `<false />` under `AllowNetworkAccess` element. Leaving the element out of `info.plist` completely is the same as having set the element to false.

3. **`Version` (optional)**
   This tag facilitates updating of the widget, allowing it to check if there is a newer version available. The `version` attribute in `config.xml` can be found in the `<widget>` element.

4. **Save the file.**

## 3. Find, Move and Rename the icon

You can find the icon file by following the path in `config.xml`:

```
<icon src="thumbnail.png"/>
```

Copy that image to the root of the widget folder. The icon must be in PNG format and named `icon.png`.

## 4. Add Javascript Libraries (optional)

If the widget uses geolocation API, copy files Bondi.txt and Platformservices.txt to the root of the widget folder. Change the extensions from `*.txt` to `*.js`.

Add the following lines inside `<head>` element in main HTML file:

(C) Copyright Nokia 2014. All rights reserved.

```
<script type="text/javascript" src="Bondi.js"></script>
<script type="text/javascript" src="Platformservices.js"></script>
```

If the widget does not use geolocation API, copy only Bondi.txt to the root of the widget folder and change the extension from `*.txt` to `*.js`.

Add the following line inside `<head>` element in main HTML file:

```
<script type="text/javascript" src="Bondi.js"></script>
```

### 5. Rezip and rename

Use a packaging tool to create a *.zip file of the widget folder. The S60 Web Runtime requires that the widget contents are in a folder inside the zip file, not as files in the root of the zip. Rename the newly created *.zip into *.wgz

### 6. Deploy

Upload the widget installation package (wgz) to your phone.

## Examples

- Bondi Widget porting example - standard Javascript
- Bondi Widget porting example - geolocation API
- Bondi Widget porting example - devicestatus API
- Extending Bondi wrapper library

## References

- Create your first WRT widget
- How to port Adobe AIR to WRT
- How to port Opera Widgets to Nokia WRT
- Web Developer's Library 1.7
- WAC SDK (former Bondi).

Note that Bondi references may require login credentials.

(C) Copyright Nokia 2014. All rights reserved.