Create your first WRT widget

09 Nov 2009

This article describes the main phases of Web Runtime (WRT) widget creation by showing how to create and deploy a very simple WRT widget.

Before you begin, make sure your development environment complies with the requirements listed in the section Before you start (WRT widget development).

Here you will learn how to create a simple widget without any specific tools, like Nokia Web Tools.

The app can be previewed quickly with a normal desktop browser, on Nokia Web Tools Simulator all Symbian WRT compatible devices or by using the Nokia Developer Remote Device Access Service.

To make more advanced Symbian WRT widgets, it's recommended to install Nokia Web Tools. In addition to providing full support for creating, editing, testing, validating packaging, and deploying widgets. The tool chain supports development for the latest WRT widget features such as homescreen and device APIs.

Creating HelloWRT

This example application generates a very basic view with some text and a simple form with text input and a button. No style sheets are used. Only very basic JavaScript™ code is used to get some interaction. When the button is clicked, the text of the input field is shown under the form as a new paragraph.

The example screenshot is 240 x 320 pixel resolution display, but the application will scale for different display sizes and orientations as well.

Creating a main HTML file

Create a new file named index.html. This is the first of two mandatory files for a WRT widget. The name of the HTML file can be anything, for example index.html. Copy and paste the following code to the index.html file and save it.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <!-- Define the title of the page -->
```


(C) Copyright Nokia 2014. All rights reserved.

```
    <title>Hello WRT!</title>
    <!-- Define the page content type -->
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <!-- Introduce the JavaScript-file the widget uses (optional) -->
    <script language="javascript" type="text/javascript" src="basic.js"></script>
    <!-- Introduce the CSS-file the widget uses (optional) -->
    <link rel="stylesheet" href="basic.css" type="text/css">
  </head>
  <body>
    <!-- Define a heading-text for the page -->
    <h1>Hello WRT!</h1>
    <!-- Create a simple form, which uses JavaScript to handle inputted text -->
    <form action="#" onsubmit="send(this.textinput1.value); return false">
      <!-- Create HTML paragraph -->
      <p>Enter your name:</p>
      <!-- Create the form input field -->
      <input name="textinput1" type=text value=""/>
      <!-- Define the form button -->
      <input type="submit" value="Say Hello!"/>
    </form>
    <!-- Define a paragraph for result text -->
    <p id="result"></p>
  </body>
</html>
```

Creating an info.plist file

Create a new file named info.plist into the same folder where index.html is created. This is the second of the two mandatory files for a WRT widget. Notice that the HTML file name inside the <string> </string> tags must match with the .html file created in step 1 (in this case index.html). Copy and paste the following lines of XML to the info.plist file and save it. When saving the files, notice that these files should be located in the root of the main folder. Since in this example we have only a few files, all files can be located into one folder and no sub folders are needed.

Notice that in this simple example, the <key>MiniViewEnabled</key> is set to "false". If it was set to "true", it would mean that the widget should have a static part to be shown on the home screen of the device. Because this HelloWRT example widget doesn't have such a part, <key>MiniViewEnabled</key> is set to "false". If you want to learn how to add the widget to the home screen of the device, please see Homescreen widget guidelines and Web Developer 's Library. Notice also that the home screen view is available only on certain Nokia devices, such as *all* Symbian^3 devices (e.g. Nokia N8), N97, N97 Mini and C6-01.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Nokia//DTD PLIST 1.0//EN" "http://www.nokia.com/DTDs/plist-
1.0.dtd">
<plist version="1.0">
<dict>
  <key>DisplayName</key>
  <string>HelloWRT</string>
  <key>Identifier</key>
  <string>com.HelloWRT.basic.widget</string>
  <key>Version</key>
  <string>1.0</string>
  <key>MainHTML</key>
  <string>index.html</string>
  <key>MiniViewEnabled</key>
  <false/>
</dict>
</plist>
```


(C) Copyright Nokia 2014. All rights reserved.

Creating a simple JavaScript file to add basic functionality

Create a new file named basic.js to the same folder as index.html, which will hold some JavaScript for the widget. This is not a mandatory file for a widget, but we have just a couple of lines of functionality for the widget there. Copy and paste the following code to the basic.js file and save it.

```
/**
 * Method to handle input data. Parameter data is appended as new paragraph
 * below the widget's form.
 * @param {Object} data The inputted text.
 */
function send(data)
{
  // Create a new paragraph for result
  var theNewParagraph = document.createElement('p');
  // Create text node for result
  var theText1 = document.createTextNode('Hello ' + data + '!');
  // Append the result text for result paragraph
  theNewParagraph.appendChild(theText1);
  // Show the result paragraph on the page
  document.getElementById("result").appendChild(theNewParagraph);
}
```

Optional step: Creating a CSS file to control style and layout

Optionally you can create a new file named basic.css, which would hold any style and layout control for the HTML elements. The file should be introduced in the index.html with the following line in the <head>-element. Note that this is not mandatory for the widget and is not used in this example.

```
<link rel="stylesheet" href="basic.css" type="text/css">
```

Example code for basic.css

```
body
{
background-color:navy;
}
```

Optional step: Adding an application icon

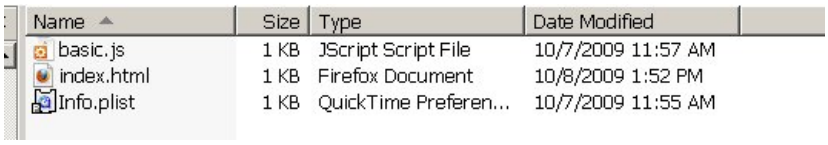
Optionally you can add an icon for the widget. If no custom icon is added, then a default application icon is shown. If you wish to add a custom application icon, it needs to be in Portable Networking Graphics (PNG) file format and named as **icon.png**. The recommended size for the icon is 88 x 88 pixels. The icon is automatically scaled, based on the device screen resolution. The custom icon needs to be in the same directory as the other widget files.

The default application icon can be seen here

Creating the widget package

Finally, create the widget package. Select the folder containing the files created above and create a ZIP file using a suitable utility, such as WinZip. Then change the extension of the resulting file to **.wgz**.

(C) Copyright Nokia 2014. All rights reserved.



Name	Size	Type	Date Modified
basic.js	1 KB	JScript Script File	10/7/2009 11:57 AM
index.html	1 KB	Firefox Document	10/8/2009 1:52 PM
Info.plist	1 KB	QuickTime Preferen...	10/7/2009 11:55 AM

Testing the widget

After you're finished coding and packaging your creation, there are several options for testing:

- Desktop browser
- Symbian WRT compatible device
- Nokia Web Tools Simulator
- Nokia Developer RDA Service

Note: Widgets do not need to be signed.

Testing the page with a desktop browser

To test the widget in a desktop browser, simply open index.html.

Running the widget on a mobile device

When the *.wgz package has been created, you can transfer it to a device like any other installable application or content.

- Deployment guide in Web Developer's Library

Running the widget in Nokia Web Tools Simulator

- Install Nokia Web Tools
- Read and watch the getting started guides

Running the widget on the Nokia Developer Remote Device Access (RDA) service

The Remote Device Access (RDA) service allows developers to test mobile applications and services remotely on various Nokia devices based on Symbian OS. It is available for all Nokia Developer members. See Nokia Developer RDA for instructions for RDA usage. Remember to delete your package file when you end the device reservation.

(C) Copyright Nokia 2014. All rights reserved.