Debugging Your First Web Application In Eclipse

For BlackBerry SmartPhones

Prosanta Bhattacherjee
Contents

Debugging Your First Web Application ........................................................................................................ 3
Introduction .................................................................................................................................................... 4
Development .................................................................................................................................................. 4
   Creating a new web site ............................................................................................................................ 4
Making Changes to the Page ......................................................................................................................... 17
Debugging your application ......................................................................................................................... 21
Debugging Windows ..................................................................................................................................... 22
Links ............................................................................................................................................................ 23
Debugging Your First Web Application

This tutorial will show you how to write a basic web application, and debug the JavaScript using the BlackBerry Web Development Plug-in for Eclipse. The application will display the GPS coordinates of a spot in the city of Waterloo.

The following is required to be able to use this tutorial:

- Windows® XP or Windows Vista™ (32-bit)
- Eclipse: Eclipse 3.4.1, EMF 2.4.1, WTP 3.0.3
- Java®: Java 1.6
- BlackBerry Web Development Plug-in for Eclipse

For help with setting up your environment please look at the tutorial How to set up Eclipse for Mobile Development found on the BlackBerry Developers Web Site (http://www.blackberry.com/developers).

If you are ready to proceed, launch Eclipse.

Figure 1
Introduction

In this tutorial, we will go over the following:

- How to create a new web site
- How to set the BlackBerry Simulator to be your default browser
- Debugging your web site using the BlackBerry Simulator

Please note that you can find more information about the BlackBerry Web Development Plug-in for Eclipse in the developer video section and the documentation found online at the BlackBerry Developer web site (http://www.blackberry.com/developers). Further documentation on Eclipse can be found through the Eclipse web site.

Development

Creating a new web site

1) From the file menu, select ‘New’ then ‘Other…’

![Figure 2]
2) Choose ‘Dynamic Web Project’ and then click ‘Next’
3) Enter the project name

![Dynamic Web Project window]

4) Click ‘Finish’
5) Your screen should now contain the project in the package Explorer

![Diagram of Package Explorer]

**Figure 5**

6) From the window menu, select ‘Show View’ and then ‘Other...’

![Screenshot of window menu with 'Other...' highlighted]

**Figure 6**
7) Extend the ‘Server’ folder, and select ‘Servers’

Figure 7
8) Right click on the servers window and select ‘New’ then ‘Server’

9) For the purposes of this demo, we’ll use an Apache web server, so expand the Apache folder and select ‘Tomcat v6.0 Server’ and then click ‘Next’

![New Server dialog box]

**Figure 8**

10) If this is the first time installing your web server, click ‘Download and Install...’ and follow the on screen prompts to install the server. You can see the progress bar in the bottom right hand corner of Eclipse. Once completed, click on ‘Browse’ and browse to the folder that Tomcat was just installed to. Then click next.

   a. If you already have a server installed, you can skip this step

**Figure 9**
11) Right click on the server and select ‘Add and Remove Projects…

12) Select the project that was just created and add it to the server and then click ‘Finish’

Figure 10
13) Right click on the WebContent folder in your Package Explorer and select ‘New’ then ‘Other’ and add an HTML page name ‘index.html’ to the folder.
14) Use the following HTML in your index.html page:

```html
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Eclipse Plug-in Tutorial</title>
  <script type="text/javascript">
    function updateDate() {
      var date = new Date();
      document.getElementById("divDate").innerHTML = date.toDateString();
    }

    function locationUpdated() { //display the new location
      var latitude = "unknown";
      var longitude = "unknown";
      var pf = navigator.platform;
      if (pf == "BlackBerry") {
        var support = blackberry.location.GPSSupported;
        if (support) { //refresh the location
          blackberry.location.refreshLocation();
          latitude = blackberry.location.latitude;
          longitude = blackberry.location.longitude;
        }
      }
      document.getElementById("latitude").innerHTML = "Latitude: " + latitude;
      document.getElementById("longitude").innerHTML = "Longitude: " + longitude;
    }
  </script>
</head>
<body onload="updateDate();">
  Hello, the current date is: <div id="divDate"></div>
  <br />
  <input type="button" value="Show Current Location" onclick="locationUpdated();" />
  <br />
  <div id="latitude" style="width:200px; height:20px;"></div>
  <div id="longitude" style="width:200px; height: 20px;"></div>
</body>
</html>

Note: You can set the GPS location of your simulator through the simulate menu on the simulator.
Right click on your server in the Servers window and select ‘Start’. This will synchronize the server with the active code.

Figure 12
16) Right click on your project in the Package Explorer and select ‘Debug As’ and then ‘BlackBerry Web’. This will bring up your Debug Configurations Window. You can name your configuration anything you’d like, and then complete the URL to include the page that you’d like to launch (in our case, add ‘index.html’ to the end of it). To begin debugging, select ‘Debug’.
Create, manage, and run configurations
Configuration for running a url of a BlackBerry web application on the simulator.

Figure 14
17) The BlackBerry simulator will now start up and launch the page you have configured. This will take a little bit of time on the first run, but for subsequent sessions, you don’t need to close the simulator, and it loads significantly faster.

Figure 15
18) We must now enable JavaScript on the simulator browser, click on the BlackBerry menu key and select ‘Options’

Figure 16
19) Then select Browser Configuration, and ensure that ‘Support JavaScript’ and ‘Allow JavaScript popups’ are selected.
20) Back out and save your changes, then select General Properties, and ensure that ‘Enable JavaScript Location support’ is checked.

![BlackBerry Simulator](image)

**Figure 18**

21) Back out and save your changes. Continue backing out until you reach the browser page.

**Making Changes to the Page**

1) You can leave the simulator running and make changes to the local copy of your webpage, and the server will automatically be updated when you save your changes.
2) Open the index.html from your Package Explorer, and change the message to read “Hello, this is my change to the html, the current date is:” and then save your changes.

3) When the server says that the ‘Status’ is ‘Synchronized’, refresh the screen on your BlackBerry Simulator and you will see your changes.

Figure 19
Debugging your application

Debugging your application is very similar to debugging any other project using Eclipse. You can debug both local and remote websites using the plug-in. To debug a remote web site you will need to point your simulator browser to that remote site and follow the same steps outlined below.

1) Open the JavaScript or html file you wish to debug from the Package Explorer

2) You can now place a breakpoint anywhere inside the JavaScript just as you would with any backend code

```
<doclet/ctag>
  <script type="text/javascript">
    function updateDate() {
       var date = new Date();
       document.getElementById("divDate").innerHTML = date.toDateString();
    }
  </script>
</doclet/ctag>
```

Figure 20
3) Once the breakpoints are placed, you’ll need to invoke the JavaScript code to hit the breakpoints and step through your code. Assuming you placed the breakpoint in the same location as the image above, you’ll just need to refresh your page on the simulator to hit the breakpoint (depending on how you have your environment set up, it may switch to debug perspective, or prompt you to. I would suggest that you switch to debug perspective, to use the full benefits of the plug-in)

![Debugging in Eclipse](image1)

**Figure 21**

4) Once a breakpoint has been hit, you have full control over the debugging environment just as you would if you were debugging backend code. All of the same hotkeys and buttons will work the same way as a normal Eclipse debug session. The debug panel looks like this:

![Debug panel](image2)

**Figure 22**

### Debugging Windows

All of the debugging windows that are normally available to you in a debug session will also be available when debugging a BlackBerry web application. For a detailed explanation of all of the
windows, please watch the video posted on the BlackBerry Developer website titled How to Use the BlackBerry Web Development Plug-in for Eclipse

Links

**BlackBerry Developers Web Site:**

http://na.blackberry.com/eng/developers/

**BlackBerry App World:**

http://na.blackberry.com/eng/developers/appworld.jsp

**BlackBerry Enterprise Server:**

http://na.blackberry.com/eng/services/server/

**BlackBerry Web Loader:**

http://www.blackberry.com/developers/downloads/weblloader/

**Developer Video Library:**

http://na.blackberry.com/eng/developers/resources/videolibrary.jsp

**Documentation:**

http://na.blackberry.com/eng/support/docs/developers/?userType=21

**Knowledge Base Articles:**


**Forums:**

http://supportforums.blackberry.com/rim/?category.id=BlackBerryDevelopment