Prior to subscribing for, installing or using any Third Party Products and Services it is your responsibility to ensure that your airtime service provider has agreed to support all of their features. Some airtime service providers may not offer Internet browsing functionality with a subscription to BlackBerry® Internet Service. Check with your service provider for availability, roaming arrangements, service plans and features. Installation or use of Third Party Products and Services with RIM's products and services may require one or more patent, trademark, copyright or other licenses in order to avoid infringement or violation of third party rights. You are solely responsible for determining whether to use, Third Party Products and Services and if any third party licenses are required to do so. If required you are responsible for acquiring them. You should not install or use Third Party Products and Services until all necessary licenses have been acquired. Any Third Party Products and Services that are provided with RIM's products and services are provided as a convenience to you and are provided "AS IS" with no express or implied conditions, endorsements, guarantees, representations or warranties of any kind by RIM and RIM assumes no liability whatsoever, in relation thereto. Your use of Third Party Products and Services shall be governed by and subject to you agreeing to the terms of separate licenses and other agreements applicable thereto with third parties, except to the extent expressly covered by a license or other agreement with RIM.

Certain features outlined in this documentation require a minimum version of BlackBerry® Enterprise Server, BlackBerry® Desktop Software, BlackBerry® Device Software and/or BlackBerry Handheld Software and may require additional development or Third Party Products and Services for access to corporate applications.

Certain products mentioned in this documentation include software developed by the Apache Software Foundation (http://www.apache.org/) and/or licensed pursuant to Apache License, Version 2.0 (http://www.apache.org/licenses/). For more information, see the NOTICE.txt file included with the software. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The terms of use of any RIM product or service are set out in a separate license or other agreement with RIM applicable thereto. NOTHING IN THIS DOCUMENTATION IS INTENDED TO SUPERSEDE ANY EXPRESS WRITTEN AGREEMENTS OR WARRANTIES PROVIDED BY RIM FOR PORTIONS OF ANY RIM PRODUCT OR SERVICE OTHER THAN THIS DOCUMENTATION.
Objectives

- Describe the features, installation, and debug capabilities of the BlackBerry® Web Plug-in for Eclipse®
- Describe the features, installation, and debug capabilities of the BlackBerry® Web Plug-in for Microsoft® Visual Studio®
- Compare Microsoft®.NET and Java® by identifying major differences
- Describe JNBridgePro® and AJAX®
- Compare and contrast Microsoft Visual Studio and Eclipse

In this chapter, you learn about web development tools for BlackBerry smartphones. These tools include the BlackBerry Web Plug-in for Eclipse and the BlackBerry Web Plug-in for Microsoft Visual Studio. This chapter also provides a comparison of the web development languages Microsoft.NET and Java and presents an overview of JNBridgePro and AJAX. You also learn about the differences between the Eclipse and Visual Studio development environments.
BlackBerry Web Plug-in for Eclipse

The BlackBerry Web Plug-in for Eclipse is a new addition to the BlackBerry web development tools. It is an extension for an existing Eclipse development environment that provides developers with tools for profiling, debugging, and testing code designed for the BlackBerry® Browser within the Eclipse environment. It enables development of web applications optimized for BlackBerry smartphones and includes a BlackBerry smartphone simulator to assist a web developer to test content. Developers can design and develop dynamic web projects using standard functionality available from within the Eclipse framework. Developers can debug and test content directly in the BlackBerry Browser using the BlackBerry smartphone simulator.

Web developers can create efficient mobile applications using existing development tools to optimize code for efficient performance, network and data connectivity, security, and compatibility with a range of BlackBerry smartphones.

The following is a list of web enhancements for the BlackBerry Web Plug-in for Eclipse.

Debugging web enhancements include the following:
- support for debugging web projects with the BlackBerry smartphone simulators
- setting breakpoints on JavaScript® code
- seamless integration into debug, variables, and expressions views

Profiling web enhancements include the following:
- visibility into the content of XMLHttpRequest object requests and response data
- visibility into data traffic for web-based content, including elements such as images, CSS, JavaScript, and HTML
- reporting on time-to-load for web-based content, including elements such as images, CSS, JavaScript, and HTML

Features of the BlackBerry Web Plug-in for Eclipse version

The BlackBerry Web Plug-in for Eclipse is a standard plug-in to Eclipse to create BlackBerry applications from within the Eclipse framework and includes the following features:
- Support for multiple BlackBerry® Device Software versions—You can change the target BlackBerry smartphone software version so that you can develop applications that make use of various APIs and BlackBerry smartphone capabilities.
- Code assist integration—You can make use of code assist that adjusts to available APIs based on the target BlackBerry smartphone software version.
- Extended Java debugging—You can debug BlackBerry applications using object, memory, and profiling views.
• Preprocessing support—You can specify preprocessing directives using fully integrated Eclipse compiler support.

• JSR integration —You can access the latest Java® ME developments using the integration of key Java Specification Requests.

• Support for existing Java libraries—You can use existing Java library projects to expand the capabilities of BlackBerry Java applications.

• Application security—You can use secure encryption and authentication for applications.

The BlackBerry Web Plug-in for Eclipse package includes the Eclipse IDE. BlackBerry® Bold™ smartphone simulator is included as the default simulator for web projects in the IDE package. The BlackBerry Web Plug-in for Eclipse requires Windows® XP or Windows Vista™ (32-bit), Eclipse version 3.4.1, EMF version 2.4.1, WTP version 3.0.3, and Java version 1.6.

Note:
Installing Eclipse Updates: North American users can use the update site to download and install the components directly into an Eclipse install. To see available components, go to the BlackBerry Developer’s web site.

BlackBerry Smartphone Simulator

The BlackBerry Smartphone Simulator includes the BlackBerry® device applications that are typically available on BlackBerry smartphones to load and test your web pages in the BlackBerry Browser. The BlackBerry Browser page in the BlackBerry Smartphone Simulator updates as you compile your code. With the simulator, you are able to do the following:

• Debug web pages—Set breakpoints, debug, and compile inline JavaScript in HTML documents. You can specify a remote URL to debug in the BlackBerry Smartphone Simulator to debug remote web pages.

• Test web pages using the BlackBerry smartphone simulator—Use the BlackBerry Smartphone Simulator to load and test web pages in the BlackBerry Browser. The BlackBerry Browser web page updates as you run the code. You can download multiple BlackBerry Smartphone Simulators for use with the tooling.

• View profiling data for project resources—Display profiling information for HTML, JavaScript, style pages, images, and other objects within your web project.

• View the contents of AJAX calls—Display HTTP request and response data generated during web page debugging.
1. Which of the following is included with the BlackBerry Web Plug-in for Eclipse? Select all that apply.
   A. Java 1.6 SDK
   B. Eclipse IDE
   C. BlackBerry smartphone simulator
   D. BlackBerry smartphone

2. What is the purpose of the BlackBerry Web Plug-in for Eclipse?
   A. Developing BlackBerry applications
   B. Debugging
   C. Profiling
   D. All of the above
   E. None of the above

3. Which of the following statements best defines the BlackBerry Web Plug-in for Eclipse?
   A. An add-on for an existing Eclipse development environment
   B. A simulation environment for debugging BlackBerry applications
   C. A tool for debugging, profiling, and developing BlackBerry applications in multiple development environments
   D. A tool for developing, debugging, and profiling BlackBerry applications in an Eclipse development environment

4. You can set breakpoints, debug, and step through inline JavaScript in HTML documents using the BlackBerry smartphone simulator. True or false?
   A. True
   B. False
5. You can use the BlackBerry Web Plug-In for Eclipse to develop, debug and test BlackBerry applications from within Eclipse and Eclipse-based IDEs. True or false?
   A. True
   B. False

6. The BlackBerry smartphone simulator must be purchased separately from the BlackBerry Web Plug-in for Eclipse. True or false?
   A. True
   B. False

7. Which of the following features of the BlackBerry Web Plug-in for Eclipse allows you to access the latest Java ME developments?
   A. Extended Java debugging
   B. Preprocessing support
   C. JSR integration
   D. Code assist integration

8. Which of the following features of the BlackBerry Web Plug-in for Eclipse allows you to use encryption and authentication for applications?
   A. Support for existing libraries
   B. Application security
   C. Code assist integration
   D. JSR integration
Answers

1. B and C
2. D
3. D
4. A
5. A
6. B
7. C
8. B
Setting up Eclipse for mobile development

Eclipse has two installations. One contains only the plug-in and requires you to install the plug-in on your development environment. The other contains both the plug-in and the Eclipse IDE. If you download the latter (plug-in and Eclipse IDE), you can skip the install steps, as everything configures for you. The following information outlines the steps to take to set up the BlackBerry Web Plug-in for Eclipse. The BlackBerry Web Plug-in for Eclipse can be found on the BlackBerry Developer web site.

Prerequisites for installing the BlackBerry Web Plug-in for Eclipse

Ensure that your computer meets the following requirements before you install and run the BlackBerry Web Plug-in for Eclipse:

- computer monitor with resolution 1024 x 768 or higher
- Intel® Pentium® 4 Processor (minimum 3 GHz)
- 1.5 GB Hard drive
- 1 GB RAM
- Microsoft Windows Vista or Windows XP
- Java® Development Kit SE version 5 or version 6 (download from the Sun® Microsystems web site). Version 6 is required if you are using BlackBerry MDS-CS for debugging.

Note:
You can use the BlackBerry Web Plug-in for Eclipse with an existing installation of Eclipse IDE for Java Developers version 3.4.0.

Installing the BlackBerry Web Plug-in for Eclipse

1. In Eclipse select Help > Software Updates.
2. Select the **Available Software** tab.

3. Click **Add Site**.

4. In the **Add Site** dialog, click **Archive**.
5. Browse to the location of the BlackBerry Web Plug-in zip file on your local machine and click **OK**.

6. Click **OK**. The BlackBerry Update Site appears in the Available Software list.

7. Select all of the items under BlackBerry Update Site.

8. If Web Tools Platform (WTP) is not installed:
   A. Verify availability of Ganymede Update Site because all dependency plug-ins download from this site.
B. In the Ganymede Update Site, under Web and Java EE Development, select Java EE Developer Tools, JavaScript Developer Tools, and Web Developer Tools.

![Software Updates and Add-ons](image)

9. Click **Install**.
   The BlackBerry Web Plug-in for Eclipse and all required dependency plug-ins are installed.

## Debugging web applications in Eclipse

This exercise demonstrates how to write a basic web application, and debug the JavaScript using the BlackBerry Web Development Plug-in for Eclipse. The application displays the GPS coordinates of a spot in the city of Waterloo. The following software is required to complete this exercise:

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

Launch Eclipse and create a web site.

1. From the **File** menu, select **New**, and then **Other**.
2. Choose **Dynamic Web Project**, and then click **Next**.

3. Enter the project name.
4. Click **Finish**.

5. Your screen now contains the project in the package Explorer.

6. From the **Window** menu, select **Show View**, and then **Other**.
7. Extend the **Server** folder, and select **Servers**.

8. Right-click on the Servers window and select **New**, and then **Server**.

9. For this demo, you can use an Apache web server. Expand the **Apache** folder and select **Tomcat v6.0 Server**, and then click **Next**.
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 corner of Eclipse. Once completed, click **Browse** and browse to the **Apache Tomcat** folder, and then click **Next**. (If you already have a server installed, you can skip this step.)

11. Right-click on the server and select **Add and Remove Projects**.

12. Select the created project and add it to the server. Click **Finish**.
13. Right-click the WebContent folder in your Package Explorer and select New, and then Other. 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>
<div id="lblUpdate">
</div>  
</body>  </html>

Note:
You can set the GPS location of your BlackBerry Smartphone Simulator through the 
Simulate menu on the simulator.

15. Right-click on your server in the Servers window and select Start. This will synchronize the server with the active code.
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, and then complete the URL to include the page you want to launch. In this case, add `index.html` to the end of it. To begin debugging, select **Debug**.
17. The BlackBerry smartphone simulator starts and launches the page you configured. This takes some time on the first run, but for subsequent sessions, you do not need to close the simulator, and it loads significantly faster.
18. You must enable JavaScript on the simulator browser. Press the BlackBerry menu key and select Options.
19. Select **Browser Configuration** and ensure that **Support JavaScript** and **Allow JavaScript pop-ups** are selected.
20. Save your changes, and then select **General Properties**. Ensure that **Enable JavaScript Location support** is checked.
21. Save your changes and close each window 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 web page, and the server updates 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 status message changes to “Synchronized”, refresh the screen on your BlackBerry Simulator and you see your changes.
Debugging your application is very similar to debugging any other project using Eclipse. You can debug both local and remote web sites using the plug-in. To debug a remote web site you need to point your simulator browser to that remote site and follow the steps outlined below.

1. Open the JavaScript or HTML file you wish to debug from the Package Explorer.

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

3. Once the breakpoints are placed, you need to invoke the JavaScript code to hit the breakpoints and run your code. You 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 do so. You may switch to Debug perspective to use the full benefits of the plug-in.
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:

**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 web site titled *How to Use the BlackBerry Web Development Plug-in for Eclipse*. 

---

38
1. You can debug both local and remote websites using the BlackBerry Web Plug-in for Eclipse. True or false?
   A. True
   B. False

2. Once a breakpoint has been hit, you have full control over the debugging environment. True or false?
   A. True
   B. False

3. Which of the following is a minimum requirement for installing the BlackBerry Web Plug-in for Eclipse 1.0?
   A. 1 GB hard drive
   B. 500 MB RAM
   C. Java SE Development Kit
   D. Eclipse IDE

4. To set up Eclipse for mobile development requires you to do which of the following?
   A. Install the plug-in
   B. Maintain the heap memory for Eclipse
   C. Disable application preprocessing
   D. Remove the plug-in

5. Which of the following is not a part of debugging a web application in Eclipse?
   A. Placing a breakpoint in JavaScript code
   B. Extending the server folder
   C. Opening the Debug Configurations Window
   D. Using the BlackBerry Smartphone simulator
6. To run an application when using BlackBerry Web Plug-in for Eclipse, in which order would you carry out the following steps? Number from 1 to 4.

A. ____Click **Activate for BlackBerry**.

B. ____Click the **Downloads** folder on the Home screen of the BlackBerry smartphone simulator, and then click the icon to open it.

C. ____Right-click the folder and click **Run As > BlackBerry Simulator**.

D. ____Open the Navigator window in Eclipse and right-click the folder containing the application.
Answers

1. A
2. A
3. C
4. A
5. B
6.
   A. 2
   B. 4
   C. 3
   D. 1
The BlackBerry Web Plug-in for Microsoft Visual Studio

The BlackBerry Web Plug-in for Microsoft Visual Studio provides you with tools for profiling, debugging, and testing code within the Microsoft Visual Studio development environment. You can use familiar tools for web development while working with features provided with the BlackBerry Web Plug-in for Visual Studio to further develop code and test it in the BlackBerry Browser.

From within Microsoft Visual Studio, you can set break points, debug, and run linked-in and inline JavaScript in ASP.NET projects. The BlackBerry Browser web page in the BlackBerry smartphone Simulator updates as you run your code. You can download and install multiple BlackBerry smartphone simulators for use with the BlackBerry Web Plug-in for Microsoft Visual Studio.

Each screen in the BlackBerry Plug-in is designed using a drag-and-drop approach to populate a form with form controls, such as buttons and text boxes. Transitions between forms can be triggered using the properties window in the UI or script. Event handling is done by creating JavaScript functions that are assigned to the control’s events in the Properties window.

The BlackBerry Web Plug-in for Microsoft Visual Studio uses JavaScript for its scripting language. The plug-in also supports Microsoft® Intellisense®, a form of automatic code completion common to the Microsoft Visual Studio development environment. When you type, Intellisense provides a list of available methods or attributes for the class or a list of parameters for the method. Not only does this minimize the amount of keystrokes required, it also limits the need to consult external documentation.

With the BlackBerry Web Plug-in for Microsoft Visual Studio, you can do the following tasks:

- Debug web pages—Set breakpoints, debug, and run inline JavaScript in HTML documents. Debug remote web pages by specifying a remote URL to debug in the BlackBerry Smartphone Simulator.
- Test web pages using the BlackBerry Smartphone Simulator—Use the BlackBerry Smartphone Simulator to load and test web pages in the BlackBerry Browser. The BlackBerry Browser web page updates as you run the code. You can download multiple BlackBerry Smartphone Simulators for use with the tooling.
- Widgets—provides you with tools for creating and packaging BlackBerry Widgets.
- View profiling data for project resources—Display profiling information for HTML, JavaScript, style pages, images, and other objects within your web project.
- View the contents of AJAX calls—Display HTTP request and response data generated during web page debugging.

The BlackBerry Web Plug-in for Microsoft Visual Studio is designed to provide a methodology similar to other applications created with Microsoft Visual Studio. The BlackBerry Plug-in makes use of the look, feel, and general design approaches used by Microsoft Visual Studio .NET and is designed to assemble applications from a set of form controls, classes and collections, and messages. The BlackBerry Web Plug-in enables you to create instances of these components, customize their properties, and use them in the application using script and graphical user interfaces.
The BlackBerry Plug-in for Microsoft Visual Studio supports the following four types of classes: user-defined, discovered, built-in, and JavaScript. It also supports collections which are classes that are encapsulated in a collection container. An instance of a collection item can be retrieved using retrieval methods for the collection. The BlackBerry Plug-in for Microsoft Visual Studio supports data collections in addition to arrays. You must use a collection when you require storage that is analogous to a database. The BlackBerry Plug-in for Microsoft Visual Studio does not support database tables.

Classes and collections include built-in Personal Information Management and JavaScript classes and collections, discovered classes and collections provided by web services, and user-defined classes and collections. Collections are containers for classes which are stored using a primary key and can be retrieved using built-in methods.

Forms are used to arrange form controls of a BlackBerry application. Each control’s events can have a script assigned to it. Button and menu item controls can also be used to trigger transitions to another form. Form controls also include labels, buttons, text boxes, masked text boxes, picture boxes, radio groups, check boxes, checked list boxes, list boxes, layout panels, date-time pickers, repeaters, and menu items.

Messages are used to communicate application data across the wireless network between the client application on the Blackberry smartphone and the data source.

**Setting up the BlackBerry Plug-in for Microsoft Visual Studio**

You need to install the necessary plugins for Microsoft Visual Studio and set up your environment to debug your client-side scripting code using the BlackBerry Plug-in for Microsoft Visual Studio. You need to run the installer for the BlackBerry Plug-in for Microsoft Visual Studio and follow the instructions in the setup wizard to complete the installation. The plug-in can be found on the BlackBerry Developer website.

**System requirements:**

- Microsoft Visual Studio Standard/Professional 2008 with Service Pack 1
- Windows XP or Windows Vista (32 bit)
- 1GHz processor or above
- 1Gb RAM or above
Configuring the BlackBerry Plug-in for Microsoft Visual Studio

Make the BlackBerry Browser the default browser. Before you can debug your web project using the BlackBerry Browser, you must configure the debugging host.

1. Right-click an .aspx page in the Solution Explorer and select Browse With.
2. In the Browse With dialog, select BlackBerry Browser.
3. Select Set as default.
4. Click Cancel. If you click Browse, the page opens in the browser, but is not in debug mode. When you start debugging, the debug session uses the BlackBerry Browser.

Displaying installed simulators

1. On the Tools menu, click Options.
2. In the left pane, expand BlackBerry. Installed simulators appear in the Simulators list.
3. To show all supported and unsupported simulators that are installed in the Show simulators for platform drop-down list, select View all Installed BlackBerry Simulators.
4. To show installed simulators for a specific BlackBerry Smartphone Software version, in the Show simulators for platform drop-down list, select the BlackBerry Smartphone Software version.
5. To refresh the list after you install additional simulators, click Refresh.

Setting the default simulator for debugging

You can change the BlackBerry Smartphone Simulator used for debugging by choosing the default simulator in the Options window.

1. On the Tools menu, click Options.
2. In the left pane, expand BlackBerry.
3. Click Browser Options.
4. In the Set default ASP.NET Simulator as drop-down list, select a simulator.
5. Click OK.
**Enabling JavaScript in the BlackBerry Browser**

1. In the BlackBerry Browser in the BlackBerry Smartphone Simulator, press the **Menu** key.
2. Click **Options**.
3. Click **Browser Configuration**.
4. Select **Support JavaScript**.
5. Select **Allow JavaScript pop-ups**.
6. Press the **Menu** key.
7. Click **Save Options**.

**Debugging your web application in Microsoft Visual Studio**

This exercise demonstrates how to write a basic web application and debug the client side script using the BlackBerry Plug-in for Microsoft Visual Studio. The application displays the GPS coordinates of a spot in the city of Waterloo. To complete this exercise, you need the following tools and computer requirements:

- MS Visual Studio Standard/Professional 2008 with Service Pack 1
- Windows XP or Windows Vista (32-bit)
- 1GHz Processor or above
- 1Gb RAM or above
- BlackBerry Plug-in for Microsoft Visual Studio

Launch Microsoft Visual Studio.

In this exercise, complete the following tasks:

- Create a new web site.
- Set the BlackBerry Simulator to be your default browser.
- Debug your web site using the BlackBerry Simulator.

**Creating a web site**

1. From the **File** menu, select **New**, and then **Web Site**.
2. Choose the type of web site you wish to create. For this exercise, begin with an empty web site.

3. Select the location of the web site. This is the location on your computer where the actual files are stored.

4. Click OK.

5. In the solution explorer, notice that a new project has been created. Right-click on that project and select Add New Item.
6. Select **HTML Page**, and then name the page. Name it index.htm, and then click **Add**.

7. On the newly created page, type “Hello, the current date is:” inside the body tags as shown below:
Hello, the current date is:

8. Set the default browser to be the BlackBerry Browser, by right-clicking on the project and selecting **Browse With**.
9. Right-click on the index.htm file in the project and select **Set As Start Page**.

10. Select BlackBerry Browser and then click **Set as Default**, and then **Browse**.
The page you created displays in the BlackBerry Simulator through the Browser. To make any changes to the page, you do not need to close the simulator. You can make your changes in the front end code and then save them. This will deploy the changes onto the Microsoft Visual Studio server, and you need only to refresh the BlackBerry Simulator browser page.

Enable JavaScript on the BlackBerry Simulators browser.

1. Push the BlackBerry menu key on the simulator and select **Options**.
2. In the **Browser Configuration** options, ensure that **Support JavaScript** and **Allow JavaScript pop-ups** are selected.
Making changes to the page

1. In the solution explorer, there are two projects. One is a proxy project and the other is your own local copy. You can leave the simulator running and make changes to your local copy. The remote project automatically updates when you save and the internal server that Visual Studio is running updates with the latest code.

2. Open the index.htm from your local copy of the project and write the front end code for the web page. In order for the sample code below to work, you need to enable JavaScript Location support which can be found in the General Properties section in the Options menu on the BlackBerry Simulator browser. Below is the sample code:

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Visual Studio 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>
<div id="lblUpdate">
</body>
</html>

---

**Note:**
You can set the GPS location of your simulator through the `Simulate` menu on the simulator.

3. When this code runs and you click the button showing the current GPS location on the simulator, it produces the results below:
Debugging your application

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

1. Open the JavaScript or html file you wish to debug from the Script Documents project.

2. Notice when you open the window that index.htm [dynamic] is open in your code view. This view of the code represents the current live view that has been pulled from the web server.

3. You can place a breakpoint anywhere inside the JavaScript just as you can with any backend code.
4. When the breakpoints are placed, you need to invoke the JavaScript code to hit the breakpoints and step through your code. You may need to refresh your page on the simulator to hit the breakpoint.

5. When you hit a breakpoint, you have full control over the debugging environment just as you do when debugging backend code. All of the same hotkeys and buttons work the same way as a normal Visual Studio debug smartphone session.

**Note:**
All of the debugging windows that are normally available to you in a debug session are also available when debugging a BlackBerry web application.
1. At what stage in the development cycle do you use the Blackberry Plug-in for Microsoft Visual Studio to create classes, collections, and forms, and then assemble them?
   A. Create the application
   B. Publish the application
   C. Implement the application
   D. Run the application

2. At what stage in the development cycle do you use the Blackberry Plug-in for Microsoft Visual Studio to deposit the application into a repository?
   A. Create the application
   B. Publish the application
   C. Implement the application
   D. Run the application

3. Which of the following is a feature of the Blackberry Plug-in for Microsoft Visual Studio?
   A. Set breakpoints
   B. Test web pages
   C. View profiling data
   D. All of the above

4. Each screen in the BlackBerry Plug-in is designed using a drag-and-drop approach. True or false?
   A. True
   B. False

5. Which one of the following items best defines BlackBerry Plug-in for Microsoft Visual Studio?
Web development tools for mobile development

A. Development environment
B. Programming language
C. Tool for debugging and testing code
D. Simulator for Microsoft Visual Studio

6. Before debugging your web project, which of the following tasks must be completed?
   A. Configure the debugging host
   B. Enable JavaScript in the BlackBerry browser.
   C. A and B
   D. None of the above

7. Which of the following items is not supported by the BlackBerry Plug-in for Microsoft Visual Studio?
   A. Database tables
   B. Data collections
   C. Arrays
   D. JavaScript classes

8. The BlackBerry Plug-in for Microsoft Visual Studio uses which of the following for its scripting languages?
   A. Perl
   B. JavaScript
   C. Python
   D. VBScript
Answers

1. A
2. B
3. D
4. A
5. C
6. C
7. A
8. B
Comparing web development languages

This section outlines the differences between Microsoft.Net and Java. It also presents a brief introduction to JNBridgePro, a tool for Java and Microsoft.Net interoperability and gives a brief overview of AJAX.

### Comparison of Microsoft.Net and Java

The following table outlines the differences between Microsoft.NET and Java:

<table>
<thead>
<tr>
<th>Microsoft.Net</th>
<th>Java</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built on C# programming language—C# language combines elements of C, C++, and Java and adds metadata tags for component development.</td>
<td>Built on Java programming language</td>
</tr>
<tr>
<td>Active Server Pages+ (ASP+)—ASP+ uses Visual Basic, C#, and other languages for code snippets. All get compiled into native code through the common language runtime. ASP+ is a new version of ASP. It supports ASPs compiling into the common language runtime.</td>
<td>Java Server Pages (JSP)—Server Pages use Java code (snippets or JavaBean references), compiled into Java bytecodes (either on-demand or batch-compiled, depending on the JSP implementation).</td>
</tr>
<tr>
<td>Runs on Microsoft Windows</td>
<td>Cross-platform</td>
</tr>
<tr>
<td>Win Forms and Web Forms—Win Forms and Web Forms RAD development are supported through the MS Visual Studio IDE.</td>
<td>Java Swing—Swing support is available in many Java IDEs and tools.</td>
</tr>
<tr>
<td>Microsoft.Net provides client-building tools to bridge Java and Microsoft.Net for application interoperability.</td>
<td>Java portability suits business rules, Java Server Pages, and Enterprise JavaBeans - allowing application code to run on different types of hardware.</td>
</tr>
</tbody>
</table>
Chapter 2

**JNBridgePro: Java and Microsoft.Net interoperability**

JNBridgePro is a tool you use to bridge Java applications to Microsoft.NET applications and vice-versa using objects and classes. You can use JNBridgePro to build applications that run over the Internet using HTTP/SOAP. JNBridgePro includes plug-ins for Eclipse and Microsoft Visual Studio.

JNBridgePro features include being able to access Java classes from Microsoft.Net and vice versa, cross-platform exception-handling, embedding widgets and control forms across platforms, and bi-directional interoperability. Java code and Microsoft.Net code can run in the same process using a shared-memory communication channel or on the same computer in different processes. Java and Microsoft.Net code can also run on different machines communicating over a network by changing only a configuration file. Java or Microsoft.Net binaries can be exposed without requiring source code.

JNBridgePro's communications architecture has a choice of three communication channels: an in-process shared-memory channel, a TCP/Binary protocol based on Microsoft.Net remoting or an HTTP/SOAP channel. JNBridgePro supports J2EE application servers.

<table>
<thead>
<tr>
<th><strong>Microsoft.Net</strong></th>
<th><strong>Java</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>ADO+ and SOAP-based Web Services—ADO+ is built on <strong>XML</strong> data interchange between remote data objects and layers of multi-tier applications on top of HTTP (SOAP). Microsoft.Net's web services tend to use SOAP messaging models.</td>
<td>JDBC, EJB, JMS and Java XML Libraries (XML4J, JAXP)—With EJB and JDBC, the developer decides the data interchange protocol, which can operate on top of either HTTP, RMI/JRMP, or IIOP.</td>
</tr>
<tr>
<td>Microsoft.Net is newer than Java and tools may not be readily available, although it allows developers to program in more than 20 languages, including Cobol; suitable for legacy environments.</td>
<td>Tools and components are more readily available.</td>
</tr>
<tr>
<td>Ease of use for building GUIs, which deliver content to web browsers; suitable for building low-cost rapidly developed applications</td>
<td>Suitable for complex applications with many business rules. It has more features for session management, fail-over, and load balancing.</td>
</tr>
<tr>
<td>Least expensive option—Microsoft.Net ships with a built-in <strong>application server.</strong></td>
<td>More expensive option—Java-based application servers are more costly per computer.</td>
</tr>
</tbody>
</table>
Using AJAX to improve your mobile web site

AJAX enhances mobile web sites by retrieving information outside a web page to update it in real time. This provides an interactive way of sharing data and promoting interoperability. This is done using the XMLHttpRequest command, which is supported by BlackBerry smartphone software version 4.6. You can use the XMLHttpRequest object to transfer data between the BlackBerry Browser and the web server without reloading the entire web page. You can use this object to send and retrieve XML, HTML, plain text, or other data types in the background. You can update the web page when new data or content is retrieved to create extremely responsive and dynamic web pages. The BlackBerry Browser supports synchronous requests (the JavaScript engine is blocked until it receives a response from the server) and asynchronous requests (the JavaScript engine can process other JavaScript functions while it awaits a response). However, because of the slower data transfer rates in a wireless browsing environment, requests made by this object should almost always be asynchronous.

The XMLHttpRequest object can help to reduce the perceived latency that marks wireless browsing. Because requests are made in the background, the user can continue to work with the current web page while new data is retrieved. Because smaller amounts of data are parsed, the BlackBerry Browser can render modifications to the web page quickly.
1. In each space provided below, write NET or J2EE to indicate which one offers the advantage stated.

A. ___ Portability tends to be good for core components such as business rules.
B. ___ This framework runs only on Windows.
C. ___ Emphasizes interoperability over portability
D. ___ Offers a more open architecture
E. ___ Supports Microsoft infrastructure
F. ___ Supports UNIX
G. ___ This environment allows developers to program in more than 20 languages.
H. ___ Tools are generally considered easier to use.
I. ___ Preferred choice for highly scalable, mission-critical applications
J. ___ Makes more sense for applications that need a low-cost, quick turnaround

2. Which of the following offers interoperability between J2EE and Microsoft.Net?

A. AJAX
B. JNBridgePro
C. CORBA IDL
D. XMLHTTP

3. Which of the following uses XMLHttpRequest to update web pages?

A. AJAX
B. JNBridgePro
C. CORBA IDL
D. DHTML
Web development tools for mobile development

Answers

1.
   A. J2EE
   B. .NET
   C. .NET
   D. J2EE
   E. .NET
   F. J2EE
   G. NET
   H. .NET
   I. J2EE
   J. .NET

2. B

3. A
A comparison of Microsoft Visual Studio and Eclipse

The following table outlines the differences between Microsoft Visual Studio and Eclipse development environments.

<table>
<thead>
<tr>
<th><strong>Eclipse</strong></th>
<th><strong>Microsoft Visual Studio</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Supports Java; supports multiple languages like Ruby and C++ with the use of plug-ins</td>
<td>Supports C/C++/Microsoft.Net</td>
</tr>
<tr>
<td>Has an open extensibility architecture with a free open source Java IDE</td>
<td>Has a closed architecture</td>
</tr>
<tr>
<td>Is available on on UNIX® and Microsoft operating systems</td>
<td>Is available on Microsoft Windows operating system</td>
</tr>
<tr>
<td>Uses a single workspace directory that contains all projects</td>
<td>Organizes files using a hierarchical structure with directories, folders, and files</td>
</tr>
<tr>
<td>Workspace is a directory in the filesystem with a .metadata subdirectory containing workspace settings and other information</td>
<td>Workspace lists projects with their dependent information such as configurations and version control</td>
</tr>
<tr>
<td>Project structure must correspond to their layout in the underlying filesystem.</td>
<td>Files do not have to correspond to filesystem folders. You can add files to a project in different directories and a reference to the new file is recorded.</td>
</tr>
<tr>
<td>Use a linked output folder that points to a directory on a local computer</td>
<td>Set the intermediate directory to a local directory</td>
</tr>
<tr>
<td>Has a local history feature that records any file changes. Deleted files can be restored from Local History.</td>
<td>Does not have a local history feature</td>
</tr>
<tr>
<td>Has multiple project types (Java and Eclipse WTP)</td>
<td>Has a single project type (C++/C#/J#)</td>
</tr>
<tr>
<td>Has an automatic build feature to determine compilation errors and runs stand-alone programs</td>
<td>Has a build command and uses a command-line structure</td>
</tr>
<tr>
<td>Uses launch configurations to collect the parameters used to launch an application. You can have separate launch configurations.</td>
<td>Generates a single executable per project and allows different launch parameters, such as command-line arguments, for different project configurations</td>
</tr>
<tr>
<td><strong>Eclipse</strong></td>
<td><strong>Microsoft Visual Studio</strong></td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>Allows multiple entry points at compile time. The entry point is determined on the command line when the program is launched.</td>
<td>Has one entry point per executable, which is determined at link time.</td>
</tr>
<tr>
<td>Has no debug mode—just the Debug perspective</td>
<td>Has a debug mode</td>
</tr>
<tr>
<td>Has a Update Manager to install updates</td>
<td>Has an Add-in Manager to install updates</td>
</tr>
<tr>
<td>Can run “eclipse.exe -clean” at the command line to rebuild most information</td>
<td>Can reset program to default state by typing “devenv/setup’ at the command prompt</td>
</tr>
<tr>
<td>Has a bug-tracking system called Eclipse Bugs, which allows you to report bugs.</td>
<td>Has a Microsoft feedback feature on the Microsoft Connect web site for customer support</td>
</tr>
</tbody>
</table>
1. Eclipse is supported by which of the following languages?
   A. Java  
   B. C++  
   C. COBOL  
   D. FORTRAN  

2. Microsoft Visual Studio is supported by which of the following languages?
   A. Java  
   B. C++  
   C. FORTRAN  
   D. COBOL  

3. Eclipse is a universal IDE with generic and customizable features. True or false?
   A. True  
   B. False  

4. Which of the following features is found in Eclipse but not in Visual Studio?
   A. Workspace  
   B. Hierarchical structure  
   C. Local history  
   D. Project directories  

5. Which of the following statements is true of Visual Studio?
   A. It uses a linked output folder that points to a directory on a local computer when building projects.
B. The structure of a project’s elements must correspond to their layout in the underlying file-system.

C. It provides a single workspace that contains all of the user’s projects.

D. It lists the projects it contains with their interdependencies, configuration, and version control.

6. Which of the following statements is/are true of Eclipse? Check all that apply.

A. It has a single project type.

B. It has multiple project types.

C. It has an automatic build feature.

D. It uses custom build steps to perform nonstandard build tasks.

7. For each of the following features, write E for Eclipse or VS for Visual Studio in the space provided to indicate to which development environment the statement applies.

A. _____It supports manually copying plug-ins into appropriate directories.

B. _____It has no debug mode, just the Debug perspective.

C. _____It has one entry point per executable.

D. _____It has an open architecture.

E. _____It records a reference to a new file to open it.

F. _____It allows you to reset everything to default state by typing devenv/setup.

G. _____Running the -clean option at the command line rebuilds most information about the installed plug-ins.
Chapter 2

Answers

1. A
2. B
3. A
4. C
5. D
6. B and C
7.
   A. E
   B. E
   C. VS
   D. E
   E. VS
   F. VS
   G. E
The BlackBerry Web Plug-in for Eclipse is an extension for an existing Eclipse development environment that provides developers with tools for profiling, debugging, and testing code designed for the BlackBerry Browser within the Eclipse environment. You can use it to develop Java applications optimized for BlackBerry smartphones and includes a BlackBerry smartphone simulator to assist a web developer to test their content.

The BlackBerry Web Plug-in for Microsoft Visual Studio provides you with tools for profiling, debugging, and testing code within the Microsoft Visual Studio development environment. You can use familiar tools for web development with this BlackBerry plug-in to further develop code and test it in the BlackBerry Browser.

When deciding between Java and Microsoft.Net as an enterprise development platform, consider application portability, existing developer skills and infrastructure, the complexity of the applications you build, vendor support, and the total economic impact that takes into account costs, benefits, and flexibility for future options. To promote Java and Microsoft.Net interoperability and improve your web site, consider the use of JNBridgePro and AJAX.

Microsoft Visual Studio and Eclipse differ as the Java programming language is different from C/C++/Microsoft.Net, and Java was the first language supported by Eclipse. The two are also different because Eclipse is a universal IDE introducing more generic and customizable features. Eclipse is also available on more operating systems. Microsoft Visual Studio offers the capability of creating secure applications in Windows Vista.
Review Questions

1. What are three key features of the BlackBerry Web Plug-in for Eclipse?

2. What are three key features of the BlackBerry Web Plug-in for Microsoft Visual Studio?

3. Based on the comparison of Eclipse and Microsoft Visual Studio, which environment would you prefer to use and why?

4. What role does each of the following tools play in the development of mobile applications?
   A. BlackBerry Web Plug-in for Eclipse
   B. BlackBerry Web Plug-in for Microsoft Visual Studio
   C. BlackBerry smartphone simulator
   D. Microsoft.Net
   E. JNBridgePro
   F. AJAX
   G. Microsoft Visual Studio
   H. Eclipse IDE