

BlackBerry Java Plug-in for Eclipse for Mac

Version: 1.1

Development Guide

Contents

1 Development Guide	3
Managing BlackBerry application projects	3
Create a BlackBerry application project	3
Create a Java source file in a BlackBerry device application project	3
Import a BlackBerry application project from an existing BlackBerry JDE workspace	3
Import a BlackBerry Java Plug-in for Eclipse sample application	4
Import a file into a BlackBerry application project	4
Delete a file from a BlackBerry application project	5
Change the properties of a BlackBerry application project	5
Project dependencies	5
Add a project dependency	5
Add a .jar file dependency	6
Specify the application settings for a BlackBerry application project	6
Specify the localization settings for a BlackBerry application project	8
Specify compiler settings for a BlackBerry Application project	9
Specify the build settings for a BlackBerry application project	10
Specify the alternate entry point settings for a BlackBerry application project	11
Managing BlackBerry device application memory	12
Locating memory allocation problems	12
Optimizing code using the Profiler	14
Building a BlackBerry device application	18
Build an application project automatically	18
Specifying preprocessor directives	18
Create preprocessor directives for a workspace	18
Create preprocessor directives for a project	19
Packaging a BlackBerry application project	19
Package a BlackBerry application project	19
Running BlackBerry device applications that use protected APIs	19
Register to use protected BlackBerry APIs	19
Using restricted code signatures	20
Request a code signature	20
Requesting a replacement registration key	21
Use a code signing key to protect packages and classes	21

Install a new code signing key.....	22
Import an existing code signing key.....	22
Remove a code signing key.....	22
View the signature status for code signature requests.....	22
Restrict access to packages or classes in a BlackBerry library project.....	23
Sign a BlackBerry application project automatically after packaging.....	24
Turn off automatic signature when you package a BlackBerry application project.....	24
Run the BlackBerry Signature Tool in the background.....	25
Suppress password prompts from the BlackBerry Signature Tool.....	25
Suppress code signing warnings.....	25
Testing and debugging BlackBerry applications.....	25
Debug a BlackBerry device application on a BlackBerry device.....	25
Test a BlackBerry device application on a BlackBerry device.....	26
Making BlackBerry device applications available to BlackBerry device users.....	26
Packaging a BlackBerry application project for multiple versions of BlackBerry Device Software.....	26
Distributing a BlackBerry device application from a web page.....	28
Distribute a BlackBerry device application from the BlackBerry Desktop Manager.....	28
Turn off .alx generation when you package a BlackBerry application project.....	28
Localizing BlackBerry device application projects.....	29
Creating resource files.....	29
Create a resource file for a BlackBerry device application.....	30
Verify the resource keys that are used in a BlackBerry device application project.....	30
Locate all of the Java files that use a resource key.....	30
Specify an original locale.....	31
Mark a translation as correct or incorrect.....	31
2 Provide feedback.....	32
3 Legal notice.....	33

Development Guide

1

Managing BlackBerry application projects

Project settings are defined in an .xml file. You can use the Eclipse® editor to change the settings.

You can import the projects that you created in the BlackBerry® JDE Plug-in for Eclipse 1.0, or in the BlackBerry® Java® Development Environment, into Eclipse 3.5 with the BlackBerry® Java® Plug-in 1.1.

Create a BlackBerry application project

When you run a BlackBerry® application project that does not contain a `public static` method, the project is treated as a library application project.

1. On the **File** menu, click **New > Project**.
2. Expand the BlackBerry® folder and select **BlackBerry Project**.
3. Click **Next**.
4. In the **Project name** field, type a name for the project.
5. Select **Create new project in workspace**.
6. Complete one of the following tasks:
 - To specify a specific JRE™, select **Use a project specific JRE**.
 - To specify the default JRE in the workspace, select **Use default JRE**.
7. Click **Next**.
8. Click **Finish**.

Create a Java source file in a BlackBerry device application project

1. In the **Package Explorer** view, expand a BlackBerry® device application project.
2. Right-click the **src** folder and select **New > Class**.
3. In the **Name** field, type a name for the .java file.
4. Click **Finish**.

Import a BlackBerry application project from an existing BlackBerry JDE workspace

You can import a project from the following development environments into Eclipse® 3.5 with the BlackBerry® Java® Plug-in for Eclipse 1.1:

- BlackBerry® Java® Development Environment.

- Eclipse with the BlackBerry® JDE Plug-in 1.0.

You cannot export a project from Eclipse 3.5 with the BlackBerry Java Plug-in for Eclipse 1.1 into an earlier version. Project files (.jdp) and workspace files (.jdw) are no longer supported.

1. On the **File** menu, click **Import**.
2. In the **Import** dialog box, expand the **BlackBerry** folder.
3. Select **Import Legacy BlackBerry Projects**.
4. Click **Next**.
5. Complete one of the following tasks:
 - To specify a specific JRE™, select **Use a project specific JRE**.
 - To specify the default JRE in the workspace, select **Use default JRE**.
6. In the **BlackBerry Workspace** field, browse to a BlackBerry workspace (.jdw) file.
7. Select the projects that you want to import.
8. Complete one of the following tasks:
 - To create a copy of the imported projects in the Eclipse workspace, select the **Copy BlackBerry projects into workspace** option.
 - To link to the original location of the imported projects, clear the **Copy BlackBerry projects into workspace** option.
9. Click **Finish**.

Import a BlackBerry Java Plug-in for Eclipse sample application

1. On the **File** menu, click **Import**.
2. In the **Import** dialog box, expand the **BlackBerry** folder.
3. Select **Import BlackBerry Samples**.
4. Click **Next**.
5. Complete one of the following tasks:
 - To specify a specific JRE™, select **Use a project specific JRE**.
 - To specify the default JRE in the workspace, select **Use default JRE**.
6. In the **BlackBerry Projects** dialog box, select the sample applications that you want to import.
7. Click **Finish**.

Import a file into a BlackBerry application project

1. In the Package Explorer view, select a BlackBerry® application project.
2. Right-click the project and click **Import**.
3. Expand the **General** folder.

4. Select the **File System** folder.
5. Click **Next**.
6. In the **From directory** dialog box, browse to the location of the source files and select the files that you want to import.
7. In the **Into folder** field, browse to the location where you want to save the files.
8. In the **Options** section, select the appropriate options.
9. Click **Finish**.

Delete a file from a BlackBerry application project

1. In the Navigator view, select a file in a BlackBerry® application project.
2. Right-click the file and click **Delete**.
3. Click **OK**.

Change the properties of a BlackBerry application project

The properties of a BlackBerry® application project are contained in the BlackBerry_App_Descriptor.xml file.

Note: The BlackBerry_App_Descriptor.xml file is not located in a source folder. A change to the BlackBerry_App_Descriptor.xml file does not trigger a Java® build and does not package a BlackBerry application project.

1. In the Package Explorer view, expand a BlackBerry application project.
2. Double-click the **BlackBerry_App_Descriptor.xml** file.
3. Change the properties and save the file.

Project dependencies

You can use project dependencies to help save time and reduce code duplication when you create a BlackBerry® device application.

Add a project dependency

1. In the Package Explorer view, right-click the project that you want to add a dependency to.
2. Click **Build Path** --> **Configure Build Path**.
3. Click the **Projects** tab.
4. Click **Add**.
5. Select the project that you want this project to depend on.
You must select either a Java® application project or a BlackBerry® application project of the Library type.
6. Click **OK**.

Add a .jar file dependency

1. In the Package Explorer view, right-click the project that you want to add a dependency to.
2. Click **Build Path > Configure Build Path**.
3. Click the **Libraries** tab.
4. Complete one of the following tasks:
 - Click **Add JARS** if the .jar file is in the current workspace.
 - Click **Add External JARS** if the .jar file is not in the current workspace.
5. Select the .jar file that you want this project to depend on.
The .jar file must be generated by a Java® application project. You cannot add a BlackBerry® application project of the Library type in this manner.
6. Click **OK**.

Specify the application settings for a BlackBerry application project

1. In the **Package Explorer** view, right-click a BlackBerry® application project and click **Properties**.
2. In the **Properties for** pane, click **BlackBerry Project**.
3. Click **Application Descriptor**.
4. Click the **Application** tab.
5. Specify the application settings.
6. On the Eclipse® toolbar, click **Save**.

Properties for application settings

Setting	Description
Title	<p>Type a descriptive name for the project.</p> <p>To create a keyboard shortcut for an application, add the Unicode underscore character (\u0332) after a letter in the title. For example, for the Hello World application, type H\u0332ello World. When the BlackBerry® device user presses the H key on the Home screen of the BlackBerry device, the Hello World application starts. If multiple applications use the same keyboard shortcut, the application that appears first on the Home screen starts.</p> <p>This setting is not available if the Application type is Library.</p>
Version	Type the version number of the project. The default value is 1.0.0.

Setting	Description
Vendor	Type the name of the company that owns the project.
Description	Type a description for the project.
Application type	<p>Select one of the following application types:</p> <ul style="list-style-type: none"> • BlackBerry Application: an application that uses CLDC/MIDP and BlackBerry APIs • MIDlet: an application that uses only MIDP APIs • Library: a library that other applications depend on <p>The default setting is BlackBerry Application.</p>
Name of main MIDlet class	<p>Type the name of the class that extends</p> <pre>javax.microedition.midlet.MIDlet</pre> <p>This field is available only if the Application type is MIDlet.</p>
Application arguments	<p>Specify the arguments to pass into the <code>main()</code> method of the application.</p> <p>This field is available only if the Application type is BlackBerry Application.</p>
Do not display the application icon on the BlackBerry home screen.	<p>Select this setting to run the application in the background, without displaying an icon on the Home screen of the BlackBerry device.</p> <p>This setting is not available if the Application type is Library.</p>
Auto-run on startup	<p>Select this setting to start the application automatically when the BlackBerry device starts.</p> <p>This setting is not available if the Application type is MIDlet.</p> <p>Applications that run automatically must be digitally signed by RIM to run on a BlackBerry device.</p>
Startup tier	<p>If the Auto-run on startup setting is selected, specify the priority in which the application is started, in relation to other applications.</p> <p>For third-party applications, you can select tier 6 or 7 (other start-up tiers are reserved for BlackBerry Applications). The default setting is 7 (lowest priority).</p> <p>This setting is not available if the Application type is MIDlet.</p>
Application icons	<p>Add the files for application icons from the project's resources or from a location external to the project.</p> <p>To specify a rollover icon, select Rollover.</p>

Setting	Description
	The number of application icons is limited to two.

Specify the localization settings for a BlackBerry application project

Before you begin:

Create the files that are required for localization. For information about localizing a BlackBerry device application, see the *BlackBerry Java Application Development Guide*.

1. In the Package Explorer view, select a BlackBerry application project.
2. Right-click the project and click **Properties**.
3. In the **Properties for** pane, select **BlackBerry Project** --> **Application Descriptor**.
4. Click the **Application** tab.
5. Specify the locale resources.
6. Click **Close**.
7. Click **Yes**.

Properties for localization settings

Setting	Description
Internationalized resource bundle available	Select this setting if the title and description of the application are internationalized in a resource bundle.
Resource bundle	Select the resource header file to use for the application (for example, HelloWorldRes).
Title ID	Select the resource key to use for the application title (for example, APPLICATION_TITLE). If you do not provide a resource key for the application title, the BlackBerry® Java® Plug-in for Eclipse® uses the title that is specified in the Title field.
Description ID	Select the resource key to use for the application description (for example, APPLICATION_DESCRIPTION). If you do not provide a resource key for the application description, the BlackBerry Java Plug-in for Eclipse uses the description that is specified in the Description field.

Specify compiler settings for a BlackBerry Application project

1. In the Navigator view, select a BlackBerry® application project.
2. Right-click the project and select **Properties**.
3. In the **Properties for** pane, select **BlackBerry Project Properties**.
4. Click the **Compile** tab.
5. Change the compiler settings.
6. Click **Apply**.
7. Click **OK**.

Properties for compiler settings

Setting	Description
Output messages about what the compiler is doing	Select this setting to view detailed compiler information when you build a project. Selecting this setting is the same as setting the <code>-verbose</code> setting for <code>rapc</code> . If you select this setting, Eclipse® will create many " <code>rapc_XXXX.dir</code> " directories in the <code>%TEMP%</code> directory during compilation.
Generate no warnings	Select this setting to hide compiler warnings when you build a project. Tip: Warnings do not prevent the project from being built successfully, but they can indicate possible errors in your code.
Treat warnings as errors	Select this setting to stop compilation if warnings occur.
Output source locations where deprecated APIs are used	Select this setting to view the location of deprecated APIs in the source files.
Don't convert image files to png	Select this setting to prevent image files from being converted to <code>.png</code> format. Select this setting if your application needs to access raw image data on the BlackBerry device.
No warning for no exported static routine: <code>.main(String[])</code>	Select this setting to prevent the compiler from issuing a warning if <code>main()</code> is undefined.
Preprocessor defines	Add, remove, or edit directives that the preprocessor uses when it processes Java® files.
Alias list	Repackage existing API libraries into a new library project. Specify one or more libraries that this project replaces, separated with semi-colons (<code>;</code>).

Setting	Description
	For example, if you are creating a library project called "newlib" that repackages existing libraries "libA" and "libB", you can specify libA;libB in this field so that .cod files that use those libraries continue to work correctly.

Specify the build settings for a BlackBerry application project

1. In the Package Explorer view, select a BlackBerry® application project.
2. Right-click the project and click **Properties**.
3. In the **Properties for** pane, select **BlackBerry Project** --> **Application Descriptor**.
4. Click the **Build** tab.
5. Specify the build settings.
6. Click **Close**.
7. Click **Yes**.

Properties for build settings

Setting	Description
Output file name	Specify the name of the .cod file that is generated when you package the project. If this field is empty, the project name is used.
Output folder	Specify the folder for the files that you create when you package a project.
Alias list	Repackage the existing API libraries into a new library project. Specify one or more libraries that this library project replaces, separated by semi-colons (;).
Preprocessor Directives	Add, delete, or change the directives that the preprocessor uses when it processes the Java® files for a project.
Output messages about what the compiler is doing	Select this setting to view detailed compiler information when you build a project.
Convert image files to .png	Select this setting to convert image files to .png format.
Create warning for no exported static routine: main(String[])	Select this setting to have the compiler issue a warning if <code>main()</code> is undefined.
Compress resources if significant space gain	Select this setting to compress the resource files.
Generate .alx file (Desktop deployment/BES Push)	Select this setting to create the application (.alx) files for distribution.

Setting	Description
Desktop Manager deployment dependencies	Add, delete, or change the external .alx files that you want to include in your application. You can use this setting to include the applications that your project depends on that are not in the current workspace. The .alx files that you add are included in <code><requires id></code> tags in the .alx file that is generated for your project.

Specify the alternate entry point settings for a BlackBerry application project

1. In the Package Explorer view, select a BlackBerry® application project.
2. Right-click the project and click **Properties**.
3. In the **Properties for** pane, select **BlackBerry Project --> Application Descriptor**.
4. Click the **Alternate Entry Points** tab.
5. Click **Add**.
6. In the **New Entry Point** dialog box, type the name and associated arguments of a project in the workspace that this project invokes.
7. Click **OK**.
8. Select the Alternate entry point properties and Locale Resources.
9. On the Eclipse® toolbar, click **Save**.

Properties for the alternate entry point settings

Setting	Description
Title	Type a name for the alternate entry point that is displayed on the Home screen of the BlackBerry® device.
Name of main MIDlet class	Type the name of the class that extends <code>javax.microedition.midlet.MIDlet</code> This field is available only if the Application type is MIDlet .
Application arguments	Specify the arguments to pass into the application's <code>main()</code> method. This field is available only if the Application type is BlackBerry application.
Do not show the application icon on the BlackBerry device Home screen	Select this setting to run the application in the background, without displaying an icon on the Home screen of the BlackBerry device. This setting is not available if the Application type is Library .

Setting	Description
Auto-run on startup	<p>Select this setting to start the application automatically when the BlackBerry device starts.</p> <p>This setting is not available if the Application type is MIDlet.</p> <p>Applications that run automatically must be digitally signed by RIM to run on a BlackBerry device.</p>
Startup tier	<p>If the Auto-run on startup setting is selected, specify the priority in which the application is started, in relation to other applications.</p> <p>For third-party applications, you can select tier 6 or 7 (other start-up tiers are reserved for BlackBerry Applications). The default setting is 7 (lowest priority).</p> <p>This setting is not available if the Application type is MIDlet .</p>
Internationalized resource bundle available	Select this setting if the title and the description of the alternate entry point have been internationalized in a resource bundle.
Resource bundle	Select the resource header file to use for the alternate entry point (for example, HelloWorldRes).
Title ID	Select the resource key to use for the application title (for example, AEP_TITLE). If you do not provide a resource key for the application title, the BlackBerry® Java® Plug-in for Eclipse® uses the title that is specified in the Title field.
Alternate Entry Point icons	<p>Add the files for application icons from the project's resources or from a location external to the project.</p> <p>To specify a rollover icon, select Rollover.</p> <p>The number of application icons is limited to two.</p>

Managing BlackBerry device application memory

Locating memory allocation problems

Memory allocation problems can cause the BlackBerry® Java® Virtual Machine to run out of flash memory, and the BlackBerry device to reset.

Viewing memory statistics

You can use the Memory Statistics tool with the Objects tool to find memory allocation problems. The Memory Statistics tool retrieves information about the memory usage of your application, and identifies the number of objects stored in memory. The Objects tool displays detailed information about each object.

Column	Description
# objects	number of objects that are currently stored in memory
Bytes in use	amount of memory that is used by objects
Allocated	total amount of memory that is allocated to the Java® Virtual Machine
Free	amount of memory that is available

Row	Description
Object handles	number of object handles that are stored in memory
RAM	RAM usage
Flash	sum of the objects that are stored in flash memory
Transient objects (flash)	number of transient objects that are stored in flash memory
Persistent objects (flash)	number of persistent objects that are stored in flash memory
Code modules (flash)	number of code modules that are stored in flash memory

View statistics to locate memory allocation problems

1. On the **Window** menu, click **Show View > Other**.
2. Expand the **BlackBerry** folder.
3. Click **BlackBerry Memory Statistics View**.
4. Click **OK**.

Display objects that are stored in memory to locate memory allocation problems

1. Specify two or more breakpoints in your code.
2. On the **Run** menu, click **Debug As > BlackBerry Device**. The application runs to the first breakpoint.
3. In the Debug view, click **Suspend**.
4. On the **Window** menu, click **Show View > Other**.
5. Expand the **BlackBerry** folder.

6. Click **BlackBerry Objects View**.
7. Click **OK**.
8. In the **BlackBerry Objects View**, click **Set up filters**.
9. In the **Objects View Options** dialog box, in the **Snapshot Filter** drop-down list, click **Compare to Snapshot**.
10. Click **OK**.
11. In the **BlackBerry Objects View**, click **Perform garbage collection**.
12. Click **Snapshot**.
13. Press **F8** to resume running the application.
14. Perform operations in the application that do not increase the number of reachable objects. For example, create a new contact and then delete it.
15. In the **Debug** view, click **Suspend**.
16. In the **BlackBerry Objects View**, click **Perform garbage collection**.
The **BlackBerry Objects View** displays the number of objects that have been deleted and added since the previous snapshot. If the number of objects added is not the same as the number of objects deleted, you might have a memory allocation problem.
17. To identify new objects, use the **Type**, **Process**, and **Location** filter types in the **Objects View Options** dialog box.
18. To save the contents of the **BlackBerry Objects View** to a comma-separated values (.csv) file, click **Save**.

Find a memory allocation problem

1. Specify two or more breakpoints in your code.
2. On the **Run** menu, click **Debug As > BlackBerry Device**. The application runs to the first breakpoint.
3. On the **Window** menu, click **Show View > Other**.
4. Expand the **BlackBerry** folder.
5. Click **BlackBerry Memory Statistics View**.
6. Click **OK**.
7. In the **BlackBerry Memory Statistics View**, click **Refresh**.
8. Click **Snapshot**.
9. Press **F8** to resume running the application. The application runs to the second breakpoint.
10. Click **Refresh**.
11. Click **Compare**.
12. Repeat steps 1 through 11, setting breakpoints closer together until they converge on the memory allocation problem.

Optimizing code using the Profiler

The profiler application collects the following information for the specified code area:

- percentage of time spent to the current point of execution.
- number objects created.
- size of objects created.
- number of objects committed.

You can improve the reliability of results and computer performance when you run the profiler application by closing other applications and increasing the heap memory in Eclipse® to 512 MB (see *Specify heap memory for Eclipse*).

Specify profiler options

1. In the **Package Explorer** view, right-click a BlackBerry® device application project.
2. Click **BlackBerry > Load Project on Device**.
3. On the **Window** menu, click **Show View > Other**.
4. Expand the **BlackBerry** folder.
5. Click **BlackBerry Profiler View**.
6. Click **OK**.
7. In the **BlackBerry Profiler View**, click **Setup Options**.

Profiler options

Drop-down list	Option	Description
Method attribution	Cumulative	The profiler application calculates the time spent executing bytecode in a method and all methods that are invoked by that method.
	In method only	The profiler application calculates the time spent executing bytecode in the specified method only. The timer stops when another method is invoked.
What to profile	Time (clock ticks)	The profiler application measures execution time (measured in clock ticks).
	Number of objects created	The profiler application calculates the number of objects that are created.
	Size of objects created	The profiler application measures the size of objects that are created.
	Number of objects committed	The profiler application calculates the number of committed objects.
	Size of objects committed	The profiler application measures the size of committed objects.

Drop-down list	Option	Description
	Number of objects moved to RAM	The profiler application calculates the number of objects that are moved into memory.
	Size of objects moved to RAM	The profiler application measures the size of objects that are moved into memory.
	User Counting	The profiler application calculates user counting.

Generate Profiler data

1. Specify a breakpoint at the beginning and end of the section of code that you want to profile.
2. On the **Run** menu, click **Debug As > BlackBerry Device**.
3. On the BlackBerry® device, run the application. The debugging tool pauses the application when it reaches the first breakpoint.
4. In Eclipse®, on the **Window** menu, click **Show View > Other**.
5. Expand the **BlackBerry** folder.
6. Click **BlackBerry Profiler View**.
7. Click **OK**.
8. In the **BlackBerry Profiler View**, click **Setup Options**.
9. In the **Profiler Options** dialog box, select the type of method attribution and the type of information that you want to profile.
10. Click **OK**.
11. Press **F8** to resume program execution.
The debugging tool pauses the application when it reaches the second breakpoint.
12. In the **BlackBerry Profiler View**, click **Refresh**.
The profiler retrieves the data. Profiler data is not cleared, so running an application again adds to the data.
13. To save the contents of the profiler pane to a comma-separated values (.csv) file, in the **BlackBerry Profiler View**, click **Save**

Viewing Profiler data

The BlackBerry Profiler Pane has three tabs. Each tab displays details about an item of execution (such as a method), the percentage of time spent running the item, and the number of times that the item was run.

Tab	Description
Summary	The Summary tab displays general statistics about the system and the garbage collector.

Tab	Description
	<p>This tab displays the percentage of time that the BlackBerry® Java® Virtual Machine has spent idle, executing code, and performing quick and full garbage collection. The Percent column displays the percentage of total BlackBerry JVM running time, including idle time and collection time.</p>
Method	<p>The Method tab displays a list of modules, sorted either by the information that you are profiling or by the number of times each item has been executed.</p> <p>On this tab, the Percent column displays the percentage of BlackBerry JVM execution time only, not including idle time and garbage collection time.</p>
Source	<p>The Source tab displays the source lines of a single method. You can navigate through the methods that invoke, and are invoked by, that method.</p> <p>You can expand a source line to show individual bytecode.</p> <p>You can further expand any bytecode that corresponds to a method invocation to show the target or targets of the method invocation.</p>

View Profiler data

1. On the **Window** menu, click **Show View > Other**.
2. Expand the **BlackBerry** folder.
3. Click **BlackBerry Profiler View**.
4. Click **OK**.

Save the contents of the BlackBerry profiler view to a .csv file

1. In the **BlackBerry Profiler View**, click **Refresh**.
2. Click **Save data to a .csv file**.
3. Select a location and type a name for the file.
4. Click **Save**.

The BlackBerry® Java® Plug-in for Eclipse® saves the contents of the BlackBerry profiler view to a comma-separated values (.csv) file. Contents are saved in the same format that they are displayed in.

Building a BlackBerry device application

Build an application project automatically

On the **Project** menu, click **Build Automatically**.

Each time you save resources in a project, an incremental build occurs.

Specifying preprocessor directives

You can specify preprocessor directives that are used to preprocess your Java® source files before they are passed to the Java compiler. You can specify preprocessor directives for a single project or for all projects in a workspace. You must turn on preprocessing in Eclipse® before you can specify preprocessor directives. For more information about how to turn on preprocessing, see the *BlackBerry Java Plug-in for Eclipse Installation and Configuration Guide*.

Directive	Description
<code>//#preprocess</code>	This directive specifies that a file must be preprocessed. This directive must be the first line in a .java file.
<code>//#ifdef tag ... #else ... #endif</code>	This directive specifies that the block before the <code>e l s e</code> is compiled if <code>t a g</code> is specified. Otherwise, only the block after the <code>e l s e</code> is compiled.
<code>//#ifndef tag ... #else ... #endif</code>	This directive specifies that the block before the <code>e l s e</code> is compiled if <code>t a g</code> is not specified. Otherwise, only the block after the <code>e l s e</code> is compiled.

If you have not turned on preprocessing, before you specify preprocessor directives you are prompted to restart Eclipse.

Create preprocessor directives for a workspace

The preprocessor directives that you create are defined for all projects in the workspace.

Before you begin: You must turn on preprocessing in Eclipse® before you can specify preprocessor directives. For more information about how to turn on preprocessing, see the *BlackBerry Java Plug-in for Eclipse Installation and Configuration Guide*.

1. On the **Eclipse** menu, click **Preferences**.
2. Expand the **BlackBerry Java Plug-in** item and click **Build**.
3. In the **Preprocessor Defines** dialog box, add, remove, or edit directives.
4. Click **OK**.

Create preprocessor directives for a project

Before you begin: You must turn on preprocessing in Eclipse® before you can specify preprocessor directives. For more information about how to turn on preprocessing, see the *BlackBerry Java Plug-in for Eclipse Installation and Configuration Guide*.

1. In the **Package Explorer** view, right-click a BlackBerry® project and click **Properties**.
2. Click **BlackBerry Project**.
3. Click **Application Descriptor**.
4. Click the **Build** tab.
5. In the **Preprocessor Directives** dialog box, type the preprocessor directives for your project.
6. On the Eclipse toolbar, click **Save**.

Packaging a BlackBerry application project

When you package a BlackBerry® application project, you create the files that you install on a BlackBerry device (for example .cod files and .alx files). When you package a BlackBerry application project, you trigger an incremental Java® build, and you also repackage the applications that depend on the BlackBerry application project.

Package a BlackBerry application project

1. In **Package Explorer** view, right-click a BlackBerry® project.
2. Click **BlackBerry > Package Project(s)**.
The **Console** view displays the progress of packaging.

Running BlackBerry device applications that use protected APIs

To use protected Research In Motion® APIs in your BlackBerry® device application, you must first receive code signing permission from RIM. For more information about code signing and to register with RIM to use protected APIs, visit www.blackberry.com/eng/developers/javaappdev/codekeys.jsp.

If a BlackBerry device application tries to access protected RIM APIs, the BlackBerry® Java® Plug-in for Eclipse® might display warning messages in the Java® editor window or in the Problems view. The warning messages appear next to the lines of application code that reference a protected API. You can suppress these warnings.

Register to use protected BlackBerry APIs

1. Visit www.blackberry.com/JDEKeys and complete the registration form.

2. Save the .csi file that Research In Motion sends to you. The .csi file contains a list of signatures and your registration information.
If the BlackBerry® Signing Authority Tool administrator does not provide you with the .csi file or the Registration PIN and you are an ISV partner, contact your ISV Technical Partnership Manager. If you are not an ISV partner, send an email message to jde@rim.com.
3. Double-click the .csi file to start the BlackBerry Signing Authority Tool .
4. If a dialog box appears that states that a private key cannot be found, follow the instructions to create a new key pair file.
5. In the **Registration PIN** field, type the PIN that RIM provided.
6. In the **Private Key Password** field, type a password of at least eight characters. The private key password protects your private key. If you lose this password, you must register again with RIM. If this password is stolen, contact RIM immediately.
7. Click **Register**.
8. Click **Exit**.

Using restricted code signatures

The BlackBerry® Signing Authority Tool administrator might place restrictions on your .csi file to limit your access to code signatures. To request changes to these restrictions, contact your administrator.

.csi file restriction	Description
# of Requests	This restriction specifies the maximum number of requests that you can make by using this .csi file. When you reach the maximum number of requests, the .csi file is no longer valid. To make new code signature requests, you must apply for a new .csi file. Although administrators can permit an infinite number of requests, they often specify a maximum number of requests for security reasons.
Expiry Date	This restriction specifies the expiry date for your .csi file. After the expiry date, you can no longer make code signature requests with this .csi file. To make new signature requests, you must apply for a new .csi file.

Request a code signature

Before you begin:

To perform this task, you must already have a .csi file that Research In Motion provides. For more information about requesting a .csi file from RIM , visit www.blackberry.com/eng/developers/javaappdev/codekeys.jsp

1. In Eclipse®, in the **Package Explorer** view, right-click a BlackBerry® application project.
2. Click **BlackBerry > Sign with Signature Tool**.
3. Select the files that require a code signature.
4. In the **Signature Tool** dialog box, click **Request**.
5. Type your private key password.
6. Click **OK**.

The BlackBerry® Signing Authority Tool uses the private key password to append the signature to the request, and sends the signed list of .cod files to the Web Signer application for verification. The Web Signer application is installed when you install the BlackBerry Signing Authority Tool . For more information about the Web Signer application, see the *BlackBerry Signing Authority Tool Version 1.0 - Password Based Administrator Guide* .

Requesting a replacement registration key

Your registration key and .csk file are stored together. If you lose the registration key or the .csk file, you cannot request code signatures.

- If you are an ISV partner and lose the .csk file, contact your ISV Technical Partnership Manager.
- If you are not an ISV partner and lose the .cskf file, send an email message to jde@rim.com.

Use a code signing key to protect packages and classes

Before you begin:

You must have the internal code signing key, the external code signing key, or both code signing keys that Research In Motion® provides. For more information about requesting code signing keys from RIM, visit www.blackberry.com/eng/developers/javaappdev/codekeys.jsp.

1. Open Eclipse®.
2. Open the project that contains the packages or classes that you want to control access to.
3. In the **Navigator** view, double-click a .key file.
4. To specify that the .key file is used to protect all public classes, select **Use as default for public classes**.
5. To specify that the .key file is used to protect all non-public classes, select **Use as default for non-public classes**.
6. Locate the name of the package that contains the packages and classes that the .key file protects.
7. Expand the package contents.
8. Select each element that requires access control.
9. Click **Ok**.

Install a new code signing key

1. On the **Eclipse** menu, click **Preferences**.
2. Expand the BlackBerry® Java® Plug-in item.
3. Click **Signature Tool**.
4. Click **Install New Keys**.
5. Navigate to the location of the signature (.csi) file that you want to add the key to and click the file.
6. Click **Open**.
7. Click **OK**.

Import an existing code signing key

1. On the **Eclipse** menu, click **Preferences**.
2. Expand the BlackBerry® Java® Plug-in item.
3. Click **Signature Tool**.
4. Click **Import Existing Keys**.
5. Navigate to the location of the BlackBerry® application project that you want to import the key from.
6. Select the folder that you want to import the key from.
7. Click **OK**.

Remove a code signing key

1. On the **Eclipse** menu, click **Preferences**.
2. Expand the BlackBerry® Java® Plug-in item.
3. Click **Signature Tool**.
4. Click **Remove Keys**.
5. Click **Yes**.
6. Click **OK**.

View the signature status for code signature requests

For files that the Web Signer application has signed, the status is Signed. For files that the Web Signer application did not sign, the status is Failed. The Web Signer application might have rejected the .cod file because the private key password was typed incorrectly.

1. Open Eclipse®.
2. On the **Project** menu, click **BlackBerry > Sign with Signature Tool**.

3. Select a .cod file.
4. View the **Status** column.

Restrict access to packages or classes in a BlackBerry library project

You can protect the packages and the classes in your BlackBerry® library project by using one or more private keys. For more information about the code signing process visit www.blackberry.com/go/devguides to see the *BlackBerry Signing Authority Tool Version 1.0 - Password Based Administrator Guide*.

Note: You can protect a package or a class with only one private key at a time.

Before you begin: You must have a key pair (public key and private key). You can get a key pair by using the BlackBerry® Signing Authority Tool. To download the BlackBerry Signing Authority Tool visit www.blackberry.com/developers/downloads.

1. Complete one of the following tasks:

Task	Steps
Copy a private key (.key) into your BlackBerry library project.	<ol style="list-style-type: none"> a. In the Package Explorer view, copy and paste the private key into the /src folder of your BlackBerry library project. b. Double-click the private key to open the key editor.
Import a private key (.key) into your BlackBerry library project.	<ol style="list-style-type: none"> a. On the File menu, click Import. b. Expand General and click File System. c. Click Next. d. In the From directory dialog box, click Browse. Navigate to the location of the private key and click OK. e. In the right-hand pane, select the private key. f. In the Into folder dialog box, click Browse and navigate to the location of the /src folder of your BlackBerry library project. g. Click the src folder and click OK. h. Click Finish. i. Double-click the private key to open the key editor.
Link a private key (.key) to your BlackBerry library project.	<ol style="list-style-type: none"> a. In the Package Explorer view, right-click the /src folder of your BlackBerry library project. b. Click New > File. c. Click Advanced.

Task	Steps
	<ol style="list-style-type: none">d. Select the Link to file in the file system option.e. Click Browse to navigate to the location of the private key and click Open.f. Click Finish. <p>The key editor opens automatically.</p>

2. In the key editor, select the package or class that you want to restrict access to.
3. Close the key editor and click **Save**.
4. Package your BlackBerry library project.
Access to the .cod file that is created is restricted at runtime.

After you finish:

When a BlackBerry device application accesses a package or a class that is protected in the BlackBerry library project, the application's .cod file must be signed with the same private key as the package or class. You can sign the .cod file with the private key by using the File Signer tool in the BlackBerry Signing Authority Tool .

Sign a BlackBerry application project automatically after packaging

1. On the **Eclipse** menu, click **Preferences**.
2. Expand the **BlackBerry Java Plug-in** item.
3. Click **Signature Tool**.
4. Select **Automatically sign the .cod files after packaging**.
5. Click **OK**.

Turn off automatic signature when you package a BlackBerry application project

By default, when you package a BlackBerry® application project, the BlackBerry® Signature Tool runs automatically to sign .cod files that access protected APIs.

To turn off this feature and run the BlackBerry Signature Tool manually, complete the following tasks:

1. On the **Eclipse** menu, click **Preferences**.
2. Expand the **BlackBerry Java Plug-in** item.
3. Click the **Signature Tool** item.
4. Clear the **Automatically sign the .cod files after packaging** option.
5. Click **OK**.

Run the BlackBerry Signature Tool in the background

You can run the BlackBerry® Signature Tool automatically after you package your BlackBerry application project.

1. On the **Eclipse** menu, click **Preferences**.
2. Expand the **BlackBerry Java Plug-in** item.
3. Click the **Signature Tool**.
4. Select the **Run Signature Tool in background** check box.
5. Click **OK**.

Suppress password prompts from the BlackBerry Signature Tool

When you package a BlackBerry® application project, the BlackBerry® Signature Tool prompts you for your signing password.

You can save your signing password for the Eclipse® session.

1. When you are prompted, in the **Signing Password** dialog box, select **Remember password**.
2. Click **OK**.

Suppress code signing warnings

1. On the **Eclipse** menu, click **Preferences**.
2. Expand the **BlackBerry Java Plug-in** item.
3. Click **Warnings**.
4. Select the keys for the protected APIs that you do not want to see warnings for.
5. Click **OK**.

Testing and debugging BlackBerry applications

Debug a BlackBerry device application on a BlackBerry device

1. Connect the BlackBerry® device to your computer using a USB cable.
2. Open the BlackBerry® Device Manager.
3. Install the application on the BlackBerry device.
4. In Eclipse®, on the **Run** menu, click **Debug Configurations**.
5. Right-click **BlackBerry Device**, and click **New**.
6. In the **Name** field, type a name.

7. Click the **BlackBerry Device** tab.
8. In the **BlackBerry Device configuration** section, perform one of the following actions:
 - Select **Attach to any connected** to attach to any BlackBerry device that is connected.
 - Select **Attach to specific device**. Select a BlackBerry device from the **Attach to BlackBerry Device** drop-down list.
9. Click **Debug**.

Test a BlackBerry device application on a BlackBerry device

1. Connect the BlackBerry device to the computer by using a USB cable.
2. In **Package Explorer** view, right-click a BlackBerry® application project.
3. Click **BlackBerry -> Load Project on Device**.

Making BlackBerry device applications available to BlackBerry device users

Packaging a BlackBerry application project for multiple versions of BlackBerry Device Software

You can create a BlackBerry® device application that is compatible with multiple versions of the BlackBerry® Device Software. When you package your BlackBerry application project (and its dependent projects) for multiple versions of the BlackBerry Device Software, the BlackBerry® Java® Plug-in for Eclipse® creates the application files for distribution to BlackBerry device users that run the versions of the BlackBerry Device Software that you include in your BlackBerry application project.

When you distribute your BlackBerry device application by using the BlackBerry® Desktop Software or the BlackBerry® Enterprise Server push service, the BlackBerry Java Plug-in for Eclipse creates one master .alx file in the Standard folder of your BlackBerry application project.

When you distribute your BlackBerry device application over the Internet, the BlackBerry Java Plug-in for Eclipse creates the .jad and .cod files for each version of the BlackBerry Device Software in the Web folder of your BlackBerry application project.

You can use the auto-signature feature in the BlackBerry® Java® Plug-in for Eclipse® to sign the .cod files automatically after you package a BlackBerry application project.

Packaging output for BlackBerry application projects

Location	File extension	Description
<project-dir>/deliverables/ Standard/	.alx	<p>You can use this file to install a BlackBerry device application on a BlackBerry device by using the BlackBerry Device Software or the BlackBerry® Enterprise Server.</p> <p>When you package a BlackBerry application project for multiple versions of the BlackBerry® Device Software, this file contains all the application files.</p> <p>You can configure the .alx file name on the Build tab of the BlackBerry_App_descriptor.xml file editor.</p>
<project-dir>/deliverables/ Standard/<x.x.x>	.cod	<p>This file is the executable file for the BlackBerry device application.</p> <p>where "x.x.x" represents the version of the BlackBerry® Java® SDK</p>
	.csi	This log file is created when you sign the .cod files in your BlackBerry device application.
	.cso	This log file contains the signature keys that are available.
	.debug	This file permits you to debug a BlackBerry device application.
	.jad	This file describes the application that is contained in the .jar file.
	.jar	This file is the application archive file that you can use to distribute your BlackBerry device application.
	.rapc	This file is necessary for packaging your BlackBerry device application.
<project-dir>/deliverables/ Web	—	The files in this location are generated only when packaging is invoked explicitly. They are intended for sharing with or copying to a deployment system.
<project-dir>/deliverables/ Web/<x.x.x>	.cod	<p>This file is the executable file for the BlackBerry device application.</p> <p>where "x.x.x" represents the version of the BlackBerry Java SDK</p>

Location	File extension	Description
	x.cod	This file is created when a large .cod file is split into one or more sibling .cod files (for example 1.cod, 2.cod, and so on).
	.csi	This log file is created when you sign the .cod files in your BlackBerry device application.
	.cso	This log file contains the signature keys that are available.
	.debug	This file permits you to debug a BlackBerry device application.
	.jad	This file describes the application that is contained in the .jar file.
	.jar	This file is the application archive file that you can use to distribute your BlackBerry device application.
	.rapc	This file is necessary for packaging your BlackBerry device application.

Distributing a BlackBerry device application from a web page

You can use the BlackBerry® Application Web Loader to allow BlackBerry device users to install a BlackBerry device application from a web page.

When you package the project by using the BlackBerry® Java® Plug-in for Eclipse®, you can distribute the .jad and .cod files that are located in the deliverables folder of the project. The application is installed on a BlackBerry device when users visit the web page that hosts the .jad and the .cod files by using the BlackBerry® Browser on the BlackBerry device.

Distribute a BlackBerry device application from the BlackBerry Desktop Manager

In the BlackBerry® Desktop Manager, you can use the application loader tool to allow BlackBerry device users to download a BlackBerry device application onto their computers and install the application on their BlackBerry devices. For more information about how to distribute a BlackBerry device application from the BlackBerry Desktop Manager, see the *BlackBerry Java Application - Core - Development Guide 5.0*. For more information about using .alx files, see the *Application Loader Online Help*.

1. In the **Package Explorer** view, expand a BlackBerry device application project.
2. Double-click the **BlackBerry_App_Descriptor.xml** file.
3. Click the **Build** tab.
4. Select **Generate .alx file (Desktop Deployment / BlackBerry Enterprise Server Push)**.
5. Distribute the .alx file and the .cod files for the BlackBerry device application to BlackBerry device users.

Turn off .alx generation when you package a BlackBerry application project

By default, when you package a BlackBerry® application project, a .alx file is generated automatically to permit you to distribute your BlackBerry device application by using the BlackBerry® Desktop Software or the BlackBerry® Enterprise Server push service.

To turn off this feature, complete the following tasks:

1. In the **Package Explorer** view, expand a BlackBerry application project.
2. Double-click the **BlackBerry_App_Descriptor.xml** file.
3. Click the **Build** tab.
4. Clear the **Generate .alx file (Desktop Deployment / BlackBerry Enterprise Server Push)** option.
5. Close the **BlackBerry_App_Descriptor.xml** file.
6. Click **Yes**.

Localizing BlackBerry device application projects

You can use the BlackBerry® Java® Plug-in for Eclipse® to localize text strings without changing the application code. The BlackBerry Java Plug-in for Eclipse provides versioning support for creating resources in different languages. You can make changes to resource values in a native language, or original locale. Translators can identify language resources that have changed and that require translation.

A `ResourceBundle` object stores information and resources for a locale. A `ResourceBundleFamily` object contains a collection of `ResourceBundle` objects. An application can use a `ResourceBundleFamily` object to display the information in a language that is specific to the locale of the BlackBerry device user.

When you compile a BlackBerry device application, the BlackBerry Java Plug-in for Eclipse compiles each `ResourceBundle` object that you include with the application into a separate code .cod file. You can install the appropriate .cod files on a BlackBerry device along with the other files that the application requires.

For more information about managing resources for localization, see the *BlackBerry Java Development Environment Fundamentals Guide*.

Creating resource files

When you create a new .rrh resource header file, the BlackBerry® Java® Plug-in for Eclipse® creates the associated .rrc resource content file. If an .rrh file already exists, the BlackBerry Java Plug-in for Eclipse creates only the new .rrc file. If an .rrh file does not already exist, the BlackBerry Java Plug-in for Eclipse creates the .rrc file and the .rrh file.

For each BlackBerry device application that you want to localize, you must create a resource header file, a resource content file for the global locale, and a resource content file for any specific locales that you require.

File	Description	Example
Resource header file	This file defines the descriptive keys for each localized string.	AppName.rrh

File	Description	Example
	During compilation, the BlackBerry Java Plug-in for Eclipse creates a resource interface automatically that uses the same name as the .rrh file with Resource appended.	
Resource content file root (global) locale	This file maps numeric keys to string literals for the root (global) locale.	AppName.rrc
Resource content file (specific locales)	This file maps descriptive keys to values for the global and specific locales (language and country). This file must have the same name as the resource header file, followed by an underscore (_) and the language code, and then, optionally, by a single underscore (_) and a country code (for example, AppName_en_GB.rrc). Two-letter language and country codes are specified in ISO-639 and ISO-3166, respectively.	AppName_en.rrc

Create a resource file for a BlackBerry device application

1. On the **File** menu, click **New > Other**.
2. In the **New** dialog box, expand the **BlackBerry** item.
3. Click **BlackBerry Resource File**.
4. Click **Next**.
5. In the **Create a new BlackBerry Resource File** dialog box, click the project folder for the BlackBerry device application.
6. In the **File name** field, type the name of the .rrc file or .rrh file.
7. Click **Finish**.

Verify the resource keys that are used in a BlackBerry device application project

1. In the **Package Explorer** view, double-click an .rrh file.
2. In the **Resource Editor** window, click **Validate**.
If one or more resource keys are not found, a message appears indicating the resource keys that are not present.

Locate all of the Java files that use a resource key

1. In the **Package Explorer** view, double-click an .rrh file.
2. In the **Resource Editor** window, click a key, and click **Validate**.

3. Click **Yes**.

If the key is found, the BlackBerry® Java® Plug-in for Eclipse® lists all of the Java® files that reference the key and the line numbers in the Java files where the key is referenced.

Specify an original locale

The original locale contains resource values in a native language. You must specify an original locale for each resource header (.rrh) file.

1. In the **Package Explorer** view, double-click an .rrh file.
2. In the **Resource Editor** window, click **Options**.
3. In the **Resource Editor Options** dialog box, select **Use versioning highlighting from given resource**.
4. In the locale drop-down list, click a locale (for example, **en**).

Mark a translation as correct or incorrect

1. In the **Package Explorer** view, double-click an .rrh file.
2. In the **Resource Editor** window, right-click a resource key.
3. Perform one of the following actions:
 - To mark a translation as correct, click **Mark Translation Correct**.
 - To mark a translation as incorrect, click **Mark Translation Incorrect**.

Provide feedback

2

To provide feedback on this deliverable, visit www.blackberry.com/docsfeedback.

Legal notice

©2010 Research In Motion Limited. All rights reserved. BlackBerry®, RIM®, Research In Motion®, SureType®, SurePress™ and related trademarks, names, and logos are the property of Research In Motion Limited and are registered and/or used in the U.S. and countries around the world.

Mac OS and Mac OS X Snow Leopard are trademarks of Apple Inc. Eclipse is a trademark of Eclipse Foundation, Inc. Intel Core and Pentium are trademarks of Intel Corporation. Perforce is a trademark of Perforce Software Incorporated. Java and Sun are trademarks of Sun Microsystems, Inc. All other trademarks are the property of their respective owners.

This documentation including all documentation incorporated by reference herein such as documentation provided or made available at www.blackberry.com/go/docs is provided or made accessible "AS IS" and "AS AVAILABLE" and without condition, endorsement, guarantee, representation, or warranty of any kind by Research In Motion Limited and its affiliated companies ("RIM") and RIM assumes no responsibility for any typographical, technical, or other inaccuracies, errors, or omissions in this documentation. In order to protect RIM proprietary and confidential information and/or trade secrets, this documentation may describe some aspects of RIM technology in generalized terms. RIM reserves the right to periodically change information that is contained in this documentation; however, RIM makes no commitment to provide any such changes, updates, enhancements, or other additions to this documentation to you in a timely manner or at all.

This documentation might contain references to third-party sources of information, hardware or software, products or services including components and content such as content protected by copyright and/or third-party web sites (collectively the "Third Party Products and Services"). RIM does not control, and is not responsible for, any Third Party Products and Services including, without limitation the content, accuracy, copyright compliance, compatibility, performance, trustworthiness, legality, decency, links, or any other aspect of Third Party Products and Services. The inclusion of a reference to Third Party Products and Services in this documentation does not imply endorsement by RIM of the Third Party Products and Services or the third party in any way.

EXCEPT TO THE EXTENT SPECIFICALLY PROHIBITED BY APPLICABLE LAW IN YOUR JURISDICTION, ALL CONDITIONS, ENDORSEMENTS, GUARANTEES, REPRESENTATIONS, OR WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY CONDITIONS, ENDORSEMENTS, GUARANTEES, REPRESENTATIONS OR WARRANTIES OF DURABILITY, FITNESS FOR A PARTICULAR PURPOSE OR USE, MERCHANTABILITY, MERCHANTABILITY, NON-INFRINGEMENT, SATISFACTORY QUALITY, OR TITLE, OR ARISING FROM A STATUTE OR CUSTOM OR A COURSE OF DEALING OR USAGE OF TRADE, OR RELATED TO THE DOCUMENTATION OR ITS USE, OR PERFORMANCE OR NON-PERFORMANCE OF ANY SOFTWARE, HARDWARE, SERVICE, OR ANY THIRD PARTY PRODUCTS AND SERVICES REFERENCED HEREIN, ARE HEREBY EXCLUDED. YOU MAY ALSO HAVE OTHER RIGHTS THAT VARY BY STATE OR PROVINCE. SOME JURISDICTIONS MAY NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES AND CONDITIONS. TO THE EXTENT PERMITTED BY LAW, ANY IMPLIED WARRANTIES OR CONDITIONS RELATING TO THE DOCUMENTATION TO THE EXTENT THEY CANNOT BE EXCLUDED AS SET OUT ABOVE, BUT CAN BE LIMITED, ARE HEREBY LIMITED TO NINETY (90) DAYS FROM THE DATE YOU FIRST ACQUIRED THE DOCUMENTATION OR THE ITEM THAT IS THE SUBJECT OF THE CLAIM.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW IN YOUR JURISDICTION, IN NO EVENT SHALL RIM BE LIABLE FOR ANY TYPE OF DAMAGES RELATED TO THIS DOCUMENTATION OR ITS USE, OR PERFORMANCE OR NON-PERFORMANCE OF ANY SOFTWARE, HARDWARE, SERVICE, OR ANY THIRD PARTY PRODUCTS AND SERVICES REFERENCED HEREIN INCLUDING WITHOUT LIMITATION ANY OF THE FOLLOWING DAMAGES: DIRECT, CONSEQUENTIAL, EXEMPLARY, INCIDENTAL, INDIRECT, SPECIAL, PUNITIVE, OR AGGRAVATED DAMAGES, DAMAGES FOR LOSS OF PROFITS OR REVENUES,

FAILURE TO REALIZE ANY EXPECTED SAVINGS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, LOSS OF BUSINESS OPPORTUNITY, OR CORRUPTION OR LOSS OF DATA, FAILURES TO TRANSMIT OR RECEIVE ANY DATA, PROBLEMS ASSOCIATED WITH ANY APPLICATIONS USED IN CONJUNCTION WITH RIM PRODUCTS OR SERVICES, DOWNTIME COSTS, LOSS OF THE USE OF RIM PRODUCTS OR SERVICES OR ANY PORTION THEREOF OR OF ANY AIRTIME SERVICES, COST OF SUBSTITUTE GOODS, COSTS OF COVER, FACILITIES OR SERVICES, COST OF CAPITAL, OR OTHER SIMILAR PECUNIARY LOSSES, WHETHER OR NOT SUCH DAMAGES WERE FORESEEN OR UNFORESEEN, AND EVEN IF RIM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW IN YOUR JURISDICTION, RIM SHALL HAVE NO OTHER OBLIGATION, DUTY, OR LIABILITY WHATSOEVER IN CONTRACT, TORT, OR OTHERWISE TO YOU INCLUDING ANY LIABILITY FOR NEGLIGENCE OR STRICT LIABILITY.

THE LIMITATIONS, EXCLUSIONS, AND DISCLAIMERS HEREIN SHALL APPLY: (A) IRRESPECTIVE OF THE NATURE OF THE CAUSE OF ACTION, DEMAND, OR ACTION BY YOU INCLUDING BUT NOT LIMITED TO BREACH OF CONTRACT, NEGLIGENCE, TORT, STRICT LIABILITY OR ANY OTHER LEGAL THEORY AND SHALL SURVIVE A FUNDAMENTAL BREACH OR BREACHES OR THE FAILURE OF THE ESSENTIAL PURPOSE OF THIS AGREEMENT OR OF ANY REMEDY CONTAINED HEREIN; AND (B) TO RIM AND ITS AFFILIATED COMPANIES, THEIR SUCCESSORS, ASSIGNS, AGENTS, SUPPLIERS (INCLUDING AIRTIME SERVICE PROVIDERS), AUTHORIZED RIM DISTRIBUTORS (ALSO INCLUDING AIRTIME SERVICE PROVIDERS) AND THEIR RESPECTIVE DIRECTORS, EMPLOYEES, AND INDEPENDENT CONTRACTORS.

IN ADDITION TO THE LIMITATIONS AND EXCLUSIONS SET OUT ABOVE, IN NO EVENT SHALL ANY DIRECTOR, EMPLOYEE, AGENT, DISTRIBUTOR, SUPPLIER, INDEPENDENT CONTRACTOR OF RIM OR ANY AFFILIATES OF RIM HAVE ANY LIABILITY ARISING FROM OR RELATED TO THE DOCUMENTATION.

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 might not offer Internet browsing functionality with a subscription to the 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, and/or BlackBerry® Device Software.

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.

Research In Motion Limited
295 Phillip Street

Waterloo, ON N2L 3W8
Canada

Research In Motion UK Limited
Centrum House
36 Station Road
Egham, Surrey TW20 9LF
United Kingdom

Published in Canada