

# BlackBerry Java Development Environment

GPS and BlackBerry Maps Development Guide

BlackBerry Java Development Environment GPS and BlackBerry Maps Development Guide

Last modified: 8 May 2008

Part number: 14793585

At the time of publication, this documentation is based on the BlackBerry Java Development Environment

Send us your comments on product documentation: <https://www.blackberry.com/DocsFeedback>.

©2008 Research In Motion Limited. All Rights Reserved. The BlackBerry and RIM families of related marks, images, and symbols are the exclusive properties of Research In Motion Limited. RIM, Research In Motion, BlackBerry, "Always On, Always Connected" and the "envelope in motion" symbol are registered with the U.S. Patent and Trademark Office and may be pending or registered in other countries.

Bluetooth is a trademark of Bluetooth SIG. Java is a trademark of Sun Microsystems, Inc. All other brands, product names, company names, trademarks and service marks are the properties of their respective owners.

The BlackBerry device and/or associated software are protected by copyright, international treaties, and various patents, including one or more of the following U.S. patents: 6,278,442; 6,271,605; 6,219,694; 6,075,470; 6,073,318; D445,428; D433,460; D416,256. Other patents are registered or pending in various countries around the world. Visit [www.rim.com/patents](http://www.rim.com/patents) for a list of RIM [as hereinafter defined] patents.

This document is provided "as is" and Research In Motion Limited and its affiliated companies ("RIM") assume no responsibility for any typographical, technical, or other inaccuracies in this document. In order to protect RIM proprietary and confidential information and/or trade secrets, this document may describe some aspects of RIM technology in generalized terms. RIM reserves the right to periodically change information that is contained in this document; however, RIM makes no commitment to provide any such changes, updates, enhancements, or other additions to this document to you in a timely manner or at all. RIM MAKES NO REPRESENTATIONS, WARRANTIES, CONDITIONS, OR COVENANTS, EITHER EXPRESS OR IMPLIED (INCLUDING WITHOUT LIMITATION, ANY EXPRESS OR IMPLIED WARRANTIES OR CONDITIONS OF FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, MERCHANTABILITY, DURABILITY, TITLE, OR RELATED TO THE PERFORMANCE OR NON-PERFORMANCE OF ANY SOFTWARE REFERENCED HEREIN OR PERFORMANCE OF ANY SERVICES REFERENCED HEREIN). IN CONNECTION WITH YOUR USE OF THIS DOCUMENTATION, NEITHER RIM NOR ITS RESPECTIVE DIRECTORS, OFFICERS, EMPLOYEES, OR CONSULTANTS SHALL BE LIABLE TO YOU FOR ANY DAMAGES WHATSOEVER BE THEY DIRECT, ECONOMIC, COMMERCIAL, SPECIAL, CONSEQUENTIAL, INCIDENTAL, EXEMPLARY, OR INDIRECT DAMAGES, EVEN IF RIM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, INCLUDING WITHOUT LIMITATION, LOSS OF BUSINESS REVENUE OR EARNINGS, LOST DATA, DAMAGES CAUSED BY DELAYS, LOST PROFITS, OR A FAILURE TO REALIZE EXPECTED SAVINGS.

This document might contain references to third-party sources of information, hardware or software, products or services and/or third-party web sites (collectively the "Third-Party Information"). RIM does not control, and is not responsible for, any Third-Party Information, including, without limitation the content, accuracy, copyright compliance, compatibility, performance, trustworthiness, legality, decency, links, or any other aspect of Third-Party Information. The inclusion of Third-Party Information in this document does not imply endorsement by RIM of the Third-Party Information or the third-party in any way. Installation and use of Third-Party Information with RIM's products and services may require one or more patent, trademark, or copyright licenses in order to avoid infringement of the intellectual property rights of others. Any dealings with Third-Party Information, including, without limitation, compliance with applicable licenses and terms and conditions, are solely between you and the third-party. You are solely responsible for determining whether such third-party licenses are required and are responsible for acquiring any such licenses relating to Third-Party Information. To the extent that such intellectual property licenses may be required, RIM expressly recommends that you do not install or use Third-Party Information until all such applicable licenses have been acquired by you or on your behalf. Your use of Third-Party Information shall be governed by and subject to you agreeing to the terms of the Third-Party Information licenses. Any Third-Party Information that is provided with RIM's products and services is provided "as is". RIM makes no representation, warranty or guarantee whatsoever in relation to the Third-Party Information and RIM assumes no liability whatsoever in relation to the Third-Party Information even if RIM has been advised of the possibility of such damages or can anticipate such damages.

Research In Motion Limited  
295 Phillip Street  
Waterloo, ON N2L 3W8  
Canada

Published in Canada

Research In Motion UK Limited  
200 Bath Road  
Slough, Berkshire SL1 3XE  
United Kingdom



# Contents

<b>1</b>	<b>Using location information in a BlackBerry device application</b>	<b>7</b>
	Getting and using GPS location information in a BlackBerry device application	7
	Using BlackBerry Maps location information in a BlackBerry device application	7
<b>2</b>	<b>Using GPS</b>	<b>9</b>
	Modes for getting GPS location information	9
	Specifying the source of GPS location information	9
	Get location information from cell site towers	10
	Get location information from GPS satellites using PDE	10
	Get location information from GPS satellites and the GPS receiver on a BlackBerry device	11
	Get GPS location information	12
	Specifying a GPS receiver as the source for location information	14
	BlackBerry smartphones with an internal GPS receiver	14
	BlackBerry smartphones with an internal GPS receiver and supporting Bluetooth GPS technology through JSR 179	15
	BlackBerry smartphones without an internal GPS receiver and supporting GPS technology through JSR 179	15
<b>3</b>	<b>Displaying location information on a BlackBerry device</b>	<b>17</b>
	Displaying location information in BlackBerry Maps	17
	Create a location document to display location information	17
	XML elements in a location document	17
	Open BlackBerry Maps from your application	18
	Open BlackBerry Maps to display the default map view	19
	Open BlackBerry Maps to display a location on a map	19
	Open BlackBerry Maps to display multiple locations on a map	20
	Open BlackBerry Maps to display a route between locations on a map	21
	Open BlackBerry Maps to display a custom map view	22
	Open BlackBerry Maps to display the location for an address in the contact list	23
	Open BlackBerry Maps from the browser	24
	Clearing data from the map that BlackBerry Maps displays	24
	Clear data specific to a location document with an id attribute	25
	Clear data specific to all location documents with an id attribute	25

Clear data specific to specific location documents with an id attribute.....	25
Clear all location data from a map.....	25
Displaying location information in your application.....	25
Configure a map in a UI field.....	26
Converting latitudinal and longitudinal coordinates to pixel values .....	26
Setting the display size of a map in your application .....	27

# Using location information in a BlackBerry device application

Getting and using GPS location information in a BlackBerry device application  
Using BlackBerry Maps location information in a BlackBerry device application

## Getting and using GPS location information in a BlackBerry device application

On a BlackBerry® device that supports Global Positioning System (GPS) technology, you can use a BlackBerry® Java® Application to display the current latitudinal and longitudinal position of a BlackBerry device. Depending on the location mode that you specify, the BlackBerry Application can also display information such as the speed and route information.

The BlackBerry Application uses the `Location` API for Java® Platform, Micro Edition (Java ME) of the `javax.microedition.location` package (JSR 179) to get location information.

The time it takes to retrieve the location of the BlackBerry device for the first time depends on several factors, such as the location mode and the GPS signal strength. In autonomous mode, typical times are less than 2 minutes. In assisted mode, typical times are less than 30 seconds.

For a BlackBerry Application on a BlackBerry device that operates on GPS networks to retrieve GPS location information, a valid SIM card must be inserted in the BlackBerry device. For more information about APIs and GPS, see the Developer Journals on the BlackBerry Developer Zone at [www.blackberry.com/developers](http://www.blackberry.com/developers). See "Modes for getting GPS location information" on page 9 for more information about using a mode to get location information.

## Using BlackBerry Maps location information in a BlackBerry device application

BlackBerry® Maps is a map and location client application that can display a map for a location, the location of the BlackBerry device, a route from a starting location to a specific ending location, and points of interest on a map.

**i Note:** BlackBerry Maps uses latitudinal and longitudinal integer values that are 100,000 times the double values specified by JSR 179, which uses WGS84 specifications for the longitudinal and latitudinal values.

A BlackBerry Application can use BlackBerry Maps in the following two ways:

- start BlackBerry Maps from a BlackBerry Java Application
- add a menu item to BlackBerry Maps that opens a BlackBerry Java Application

BlackBerry Maps is included on BlackBerry devices that run BlackBerry® Device Software version 4.2 and later. To download BlackBerry Maps to a BlackBerry device that runs BlackBerry Device Software version 4.1, visit [www.blackberry.com/maps](http://www.blackberry.com/maps). See "Displaying location information in BlackBerry Maps" on page 17 for more information.

# Using GPS

Modes for getting GPS location information

Specifying the source of GPS location information

Specifying a GPS receiver as the source for location information

## Modes for getting GPS location information

To specify the location mode that a BlackBerry® device application uses to get location information, you can create the `Criteria` object and pass it as a parameter to `LocationProvider.getInstance()`.

Location mode	Description
cell site	<p>Get location information from the location of cell site towers and strength of signal</p> <p><b>Advantages</b></p> <ul style="list-style-type: none"> <li>the fastest location mode</li> </ul> <p><b>Disadvantages</b></p> <ul style="list-style-type: none"> <li>accuracy of location information is low-level and does not provide tracking information such as speed or route information</li> </ul> <p><b>Requirements</b></p> <ul style="list-style-type: none"> <li>both the BlackBerry device and the wireless service provider must support this mode</li> <li>wireless network coverage</li> </ul>
assisted	<p>Get location information from satellites using a Position Determination Entity (PDE)</p> <p><b>Advantages</b></p> <ul style="list-style-type: none"> <li>provides location information faster than the autonomous mode and more accurately than the cell site mode</li> </ul> <p><b>Requirements</b></p> <ul style="list-style-type: none"> <li>both the BlackBerry device and the wireless service provider must support this functionality</li> <li>wireless network coverage</li> </ul>
autonomous	<p>Get location information from the GPS receiver on the BlackBerry device without assistance from the wireless network</p> <p><b>Advantages</b></p> <ul style="list-style-type: none"> <li>accuracy of location information is high-level</li> <li>does not require assistance from the wireless network</li> </ul> <p><b>Disadvantages</b></p> <ul style="list-style-type: none"> <li>slower than other modes</li> </ul>

## Specifying the source of GPS location information

A BlackBerry® Application selects the source of location information depending on how closely the GPS location provider matches the defined criteria.

- > To specify the mode for getting GPS location information, create an instance of the `javax.microedition.location.Criteria` class, invoke the appropriate set methods, and then pass the instance to `LocationProvider.getInstance()`.

## Get location information from cell site towers

1. Create an instance of a `Criteria` object.  

```
Criteria criteria = new Criteria();
```
2. Perform the following tasks:

Task	Steps
Specify longitudinal accuracy is not required.	> Invoke <code>criteria.setHorizontalAccuracy(NO_REQUIREMENT)</code> . <code>criteria.setHorizontalAccuracy(NO_REQUIREMENT);</code>
Specify latitudinal accuracy is not required.	> Invoke <code>criteria.setVerticalAccuracy(NO_REQUIREMENT)</code> . <code>criteria.setVerticalAccuracy(NO_REQUIREMENT);</code>
Specify that this mode can incur cost.	> Invoke <code>criteria.setCostAllowed(true)</code> . <code>criteria.setCostAllowed(true);</code>
Specify power consumption is low.	> Invoke <code>criteria.setPreferredPowerConsumption(POWER_USAGE_LOW)</code> . <code>setPreferredPowerConsumption(POWER_USAGE_LOW);</code>

3. Invoke `LocationProvider.getInstance()`, storing the returned object in a `LocationProvider` object.  

```
LocationProvider provider = LocationProvider.getInstance(criteria);
```

## Get location information from GPS satellites using PDE

1. Create an instance of a `Criteria` object.  

```
Criteria criteria = new Criteria();
```
2. Perform the following tasks:

Task	Steps
Specify longitudinal accuracy is not required.	> Invoke <code>criteria.setHorizontalAccuracy(NO_REQUIREMENT)</code> . <code>criteria.setHorizontalAccuracy(NO_REQUIREMENT);</code>
Specify latitudinal accuracy is not required.	> Invoke <code>criteria.setVerticalAccuracy(NO_REQUIREMENT)</code> . <code>criteria.setVerticalAccuracy(NO_REQUIREMENT);</code>
Specify that this mode can incur cost.	> Invoke <code>criteria.setCostAllowed(true)</code> . <code>criteria.setCostAllowed(true);</code>
Specify power consumption is medium or not required.	> Invoke <code>criteria.setPreferredPowerConsumption(int level)</code> using one of <code>POWER_USAGE_MEDIUM</code> , or <code>NO_REQUIREMENT</code> as the parameter. <code>criteria.setPreferredPowerConsumption(POWER_USAGE_MEDIUM);</code>

3. Invoke `LocationProvider.getInstance()`, storing the returned object in a `LocationProvider` object.  

```
LocationProvider provider = LocationProvider.getInstance(criteria);
```

## Get location information from GPS satellites and the GPS receiver on a BlackBerry device

1. Create an instance of a `Criteria` object.  
`Criteria criteria = new Criteria();`
2. Perform one of the following sets of tasks:

### Get accurate location information

Task	Steps
Specify longitudinal accuracy is required.	> Invoke <code>criteria.setHorizontalAccuracy(int accuracy)</code> using an integer value that represents the accuracy, in meters. <code>criteria.setHorizontalAccuracy(50);</code>
Specify latitudinal accuracy is required.	> Invoke <code>criteria.setVerticalAccuracy(int accuracy)</code> using an integer value that represents the accuracy, in meters. <code>criteria.setVerticalAccuracy(50);</code>

### Get approximate location information

Task	Steps
Specify longitudinal accuracy is not required.	> Invoke <code>criteria.setHorizontalAccuracy(NO_REQUIREMENT)</code> . <code>criteria.setHorizontalAccuracy(NO_REQUIREMENT);</code>
Specify latitudinal accuracy is not required.	> Invoke <code>criteria.setVerticalAccuracy(NO_REQUIREMENT)</code> . <code>criteria.setVerticalAccuracy(NO_REQUIREMENT);</code>

3. To specify that this mode can incur cost, invoke `criteria.setCostAllowed(true)`.
4. To specify power consumption is high, invoke `criteria.setPreferredPowerConsumption(POWER_USAGE_HIGH)`.
5. Invoke `LocationProvider.getInstance()`, storing the returned object in a `LocationProvider` object.  
`LocationProvider provider = LocationProvider.getInstance(criteria);`

## Get location information from the GPS receiver on a BlackBerry device

1. Create an instance of a `Criteria` object.  
`Criteria criteria = new Criteria();`

## 2. Perform one of the following sets of tasks:

Task	Steps
Get accurate location information using the GPS receiver on a BlackBerry device with no cost and no power consumption.	<ol style="list-style-type: none"> <li>1. Invoke <code>criteria.setHorizontalAccuracy(int accuracy)</code> using an integer value that represents the accuracy, in meters. <code>criteria.setHorizontalAccuracy(50);</code></li> <li>2. Invoke <code>criteria.setVerticalAccuracy(int accuracy)</code> using an integer value that represents the accuracy, in meters. <code>criteria.setVerticalAccuracy(50);</code></li> <li>3. Invoke <code>criteria.setCostAllowed(false)</code>. <code>criteria.setCostAllowed(false);</code></li> </ol>
Get accurate location information using the GPS receiver on a BlackBerry device with cost and variable power consumption.	<ol style="list-style-type: none"> <li>1. Invoke <code>criteria.setHorizontalAccuracy(int accuracy)</code> using an integer value that represents the accuracy, in meters. <code>criteria.setHorizontalAccuracy(50);</code></li> <li>2. Invoke <code>criteria.setVerticalAccuracy(int accuracy)</code> using an integer value that represents the accuracy, in meters. <code>criteria.setVerticalAccuracy(50);</code></li> <li>3. Invoke <code>criteria.setCostAllowed(true)</code>. <code>criteria.setCostAllowed(true);</code></li> <li>4. Invoke <code>criteria.setPreferredPowerConsumption(int level)</code> using one of <code>POWER_USAGE_LOW</code>, <code>POWER_USAGE_MEDIUM</code>, or <code>NO_REQUIREMENT</code> as the parameter. <code>criteria.setPreferredPowerConsumption(POWER_USAGE_MEDIUM);</code></li> </ol>
Get approximate location information using the GPS receiver on a BlackBerry device with no cost and variable power consumption.	<ol style="list-style-type: none"> <li>1. Invoke <code>criteria.setHorizontalAccuracy(NO_REQUIREMENT)</code>. <code>criteria.setHorizontalAccuracy(NO_REQUIREMENT);</code></li> <li>2. Invoke <code>criteria.setVerticalAccuracy(NO_REQUIREMENT)</code>. <code>criteria.setVerticalAccuracy(NO_REQUIREMENT);</code></li> <li>3. Invoke <code>criteria.setCostAllowed(false)</code>. <code>criteria.setCostAllowed(false);</code></li> <li>4. Invoke <code>criteria.setPreferredPowerConsumption(int level)</code> using one of <code>POWER_USAGE_MEDIUM</code>, <code>POWER_USAGE_HIGH</code>, or <code>NO_REQUIREMENT</code> as the parameter. <code>criteria.setPreferredPowerConsumption(POWER_USAGE_MEDIUM);</code></li> </ol>

3. Invoke `LocationProvider.getInstance()`, storing the returned object in a `LocationProvider` object.

```
LocationProvider provider = LocationProvider.getInstance(criteria);
```

## Get GPS location information

GPS location information might be unavailable if the BlackBerry® device cannot receive signals from GPS satellites. This situation can occur when something is obstructing the signals, such as buildings or dense clouds.

Task	Steps
Specify a response time for getting the location of the BlackBerry device.	> Invoke <code>Criteria.setPreferredResponseTime()</code> , and specify the response time, in milliseconds.

Task	Steps
Get the location information for a BlackBerry device.	<p>&gt; In a non-event thread, invoke <code>LocationProvider.getLocation(int)</code>, providing a timeout, in seconds.</p> <pre>try { // Specify -1 to have the implementation use its default timeout value // for this provider. Location location = provider.getLocation(-1); } catch (Exception e) { // handle LocationException, InterruptedException, SecurityException // and IllegalArgumentException }</pre>
Get the speed of the terminal, in meters per second.	<p>&gt; Invoke <code>location.getSpeed()</code>.</p> <pre>float speed = location.getSpeed();</pre>
Get the course of the terminal, in degrees.	<p>&gt; Invoke <code>location.getCourse()</code>.</p> <pre>float course = location.getCourse();</pre>
Get the number of GPS satellites that a BlackBerry device tracks.	<ol style="list-style-type: none"> <li>1. Invoke the <code>Location.getExtraInfo()</code> method using the National Marine Electronics Association (NMEA) MIME type as a parameter. The method returns an encoded <code>String</code> that contains information on the number of GPS satellites. <pre>String encodedStr = Location.getExtraInfo("application/X-jsr179-location-nmea");</pre> </li> <li>2. To retrieve the GPS satellite information, parse through the information in the returned <code>String</code> value.</li> </ol> <p>See the <i>API reference</i> for the BlackBerry® Java® Development Environment for more information about the <code>Location.getExtraInfo()</code> method.</p>
Get the time that a BlackBerry device application retrieved GPS information.	<p>&gt; Invoke <code>Location.getTimeStamp()</code>. The time is retrieved from the GPS receiver.</p> <pre>Location.getTimeStamp();</pre>
Maintain a connection with GPS satellites.	<p>Configure a BlackBerry device application to query the Location API for GPS location information in an interval less than 10 seconds.</p> <p>&gt; Define a <code>LocationListener</code> with an interval value that is less than 10 seconds.</p> <pre>LocationProvider provider = LocationProvider.getInstance(criteria); provider.setLocationListener(loclistener, 9, -1, -1);</pre>
Get continuous GPS readings using the cell site mode.	<p>&gt; Use the <code>LocationProvider.getLocation()</code> method with a timer to simulate the retrieval of GPS information at fixed intervals.</p>
Get GPS information during a phone call.	<p>&gt; Use the autonomous mode to retrieve GPS information. See "Get location information from the GPS receiver on a BlackBerry device" on page 11 for more information.</p>
Get location position updates.	<p>You can associate only one <code>LocationListener</code> with a particular location provider for GPS location information. BlackBerry Java Applications typically listen for updates on a separate thread.</p> <ol style="list-style-type: none"> <li>1. Implement the <code>LocationListener</code> interface.</li> <li>2. To register your implementation, invoke <code>LocationProvider.setLocationListener()</code>.</li> </ol>

Task	Steps
Change the criteria for a LocationProvider that is associated with an LocationListener.	<ol style="list-style-type: none"> <li>1. Create a new Criteria object. <code>Criteria criteria2 = new Criteria();</code></li> <li>2. Set the new criteria by invoking the <code>setHorizontalAccuracy()</code>, <code>setVerticalAccuracy()</code>, <code>setCostAllowed()</code>, and <code>setPreferredPowerConsumption()</code> methods of the new Criteria object.</li> <li>3. Invoke <code>LocationProvider.reset()</code>. <code>LocationProvider.reset();</code></li> <li>4. Set the LocationListener to null. <code>provider.setLocationListener(null, -1, -1, -1);</code></li> <li>5. Create a new instance of a LocationProvider by invoking <code>LocationProvider.getInstance()</code> using the new Criteria object as a parameter. <code>provider = LocationProvider.getInstance(criteria2);</code></li> <li>6. Set the LocationListener. <code>provider.setLocationListener(LocationListener listener, int interval, int timeout, int maxAge);</code></li> </ol>

## Specifying a GPS receiver as the source for location information

On a BlackBerry® device with BlackBerry® Device Software version 4.2 or later, you can configure a GPS receiver that uses Bluetooth® technology as the data source for location information that the Location API uses when a BlackBerry device application accesses the Location API. You can also configure the frequency that a BlackBerry device uses to request location information from a Bluetooth enabled GPS receiver.

If only one Bluetooth enabled GPS receiver is paired with a BlackBerry device, the Location API will try to use the Bluetooth enabled GPS receiver as the source for location information.

If more than one Bluetooth enabled GPS receiver is paired with a BlackBerry device, the BlackBerry device user can select the Bluetooth enabled GPS receiver as the source for location information for the Location API.

- On a BlackBerry device with BlackBerry Device Software version 4.2 or later that does not have an internal GPS receiver, a BlackBerry device user can specify one or more Bluetooth enabled GPS receivers as the data source for location information that the Location API uses. When a BlackBerry device application attempts to access the Location API, and GPS is unavailable on the BlackBerry device, the BlackBerry device will automatically connect to and use the Bluetooth enabled GPS receiver as the data source for location information that the Location API uses.
- On a BlackBerry device with BlackBerry Device Software version 4.2 or later with an internal GPS receiver, a BlackBerry device user can choose between internal GPS or a Bluetooth enabled GPS receiver as the data source for location information that the Location API uses.

### BlackBerry smartphones with an internal GPS receiver

- BlackBerry® 7100i smartphone
- BlackBerry® 7130e smartphone

- BlackBerry® 7250 smartphone
- BlackBerry® 7520 smartphone
- BlackBerry® 8703e smartphone

### **BlackBerry smartphones with an internal GPS receiver and supporting Bluetooth GPS technology through JSR 179**

- BlackBerry® Pearl™ 8110 smartphone
- BlackBerry® Pearl™ 8130 smartphone
- BlackBerry® Curve™ 8310 smartphone
- BlackBerry® 8800 smartphone
- BlackBerry® 8820 smartphone
- BlackBerry® 8830 World Edition smartphone

### **BlackBerry smartphones without an internal GPS receiver and supporting GPS technology through JSR 179**

- BlackBerry® 8100 smartphone
- BlackBerry® Pearl™ 8120 smartphone
- BlackBerry® 8300 smartphone
- BlackBerry® Curve™ 8320 smartphone
- BlackBerry® 8700 Series



# Displaying location information on a BlackBerry device

- Displaying location information in BlackBerry Maps
- Open BlackBerry Maps from your application
- Open BlackBerry Maps from the browser
- Clearing data from the map that BlackBerry Maps displays
- Displaying location information in your application

## Displaying location information in BlackBerry Maps

The BlackBerry® Maps uses a location document to display location information on a map, such as locations and routes. A location document is a `String` that contains a set of XML elements with attributes that specify information about a location or route. You can include information about ten locations in one location document.

### Create a location document to display location information

> Perform one of the following actions:

Task	Steps
Use required elements only.	<ul style="list-style-type: none"><li>&gt; Create a string using the following format:<pre>"&lt;lbs&gt; &lt;location y='latitude' x='longitude' label='Location_Label' description='Description' /&gt; &lt;location y='latitude' x='longitude' label='Location_Label' description='Description' /&gt; &lt;location y='latitude' x='longitude' label='Location_Label' description='Description' /&gt; ...&lt;/lbs&gt;"</pre></li></ul>
Use required and optional elements.	<ul style="list-style-type: none"><li>&gt; Create a string using the following format:<pre>"&lt;lbs&gt; &lt;location y='latitude' x='longitude' label='Location_Label' description='Description' zoom='Zoom' address='address' city='city' region='region' country='country' postalCode='postal_code' phone='phone' fax='fax' url='url' email='email' categories='categories' rating='rating' /&gt; &lt;/lbs&gt;"</pre></li></ul>

## XML elements in a location document

### The `<lbs>` element

The `<lbs>` and `</lbs>` elements encapsulate the information in a location document. The opening and closing `<lbs>` elements contain all other location document elements. The `<lbs>` element has the following optional attributes:

Attribute	Type	Description	Required	BlackBerry Java Development Environment version
id	String	the id of a location document - cannot be NONE, ,DOCS, LOCATIONS, ROUTES, or ALL	no	4.5.0 or later
clear	String	NONE   DOCS   LOCATIONS   ROUTES   ALL   <i>String, String, ..., String</i>	no	4.5.0 or later

## The `<getRoute>` element

The `<getRoute>` and `</getRoute>` elements contain route information. To display route information in a map, you must place two `<location>` elements within the opening and closing `<getRoute>` elements. The first `<location>` element represents the starting point of a route while the second `<location>` element represents the ending point of a route. You can only use the x and y attributes in a `<location>` element that is nested within a `<getRoute>` element.

## The `<location>` element

The `<location>` element contains information for a specific location and has the following required and optional attributes:

Attribute	Type	Description	Required	BlackBerry Java Development Environment version
y	integer	latitude in decimal degrees x 100,000	yes	4.2.0 or later
x	integer	longitude in decimal degrees x 100,000	yes	4.2.0 or later
zoom	integer	zoom level from 0 to MAX_ZOOM	yes	4.2.0 or later
label	string	a label that is displayed beside a location on a map	yes	4.2.0 or later
description	string	description information for a location	yes	4.2.0 or later
address	string	address	no	4.2.1 or later
city	string	city	no	4.2.1 or later
region	string	province or state	no	4.2.1 or later
country	string	country	no	4.2.1 or later
postalCode	string	postal code	no	4.2.1 or later
phone	string	phone number	no	4.2.1 or later
fax	string	fax number	no	4.2.1 or later
url	string	URL	no	4.2.1 or later
email	string	email address	no	4.2.1 or later
categories	string	category	no	4.2.1 or later
rating	double	rating information, between 0 and 5	no	4.2.1 or later

## Open BlackBerry Maps from your application

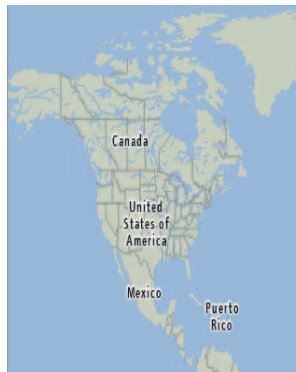
- > Your application must import the `Invoke` and the `MapsArguments` classes.

## Open BlackBerry Maps to display the default map view

- > To display the last map that the BlackBerry® device user viewed, from a BlackBerry Application, invoke `invokeApplication()` using a new `MapsArguments` object that does not contain any arguments, for example, `Invoke.invokeApplication( Invoke.APP_TYPE_MAPS, new MapsArguments() );`

### View a sample application that displays the default map

1. From the BlackBerry® Integrated Development Environment, start the **BlackBerryMapsDemo** project.
2. In the BlackBerry Smartphone Simulator, click **BlackBerry Maps Demo**.
3. Open the menu and click **Invoke Default**.
4. Open the menu and click **View Map**. BlackBerry® Maps displays the following map:



## Open BlackBerry Maps to display a location on a map

1. Create a `String` that contains location information for one location, for example  

```
String document = "<lbs><location lon='-7938675' lat='4367022' label='Toronto, ON' description='Toronto' zoom='10' /></lbs>";
```
2. Invoke `invokeApplication()` using the `APP_TYPE_MAPS` constant parameter, a new `MapsArguments` object that uses the `ARG_LOCATION_DOCUMENT` property and the location `String`, for example  

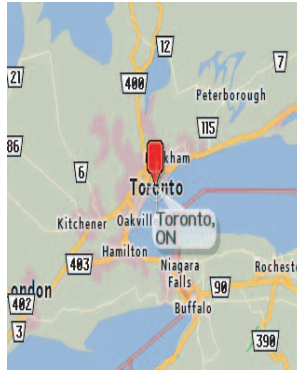
```
Invoke.invokeApplication(Invoke.APP_TYPE_MAPS, new MapsArguments(  
MapsArguments.ARG_LOCATION_DOCUMENT, document));
```

### View a sample application that displays a location on a map

1. From the BlackBerry® Integrated Development Environment, start the **BlackBerryMapsDemo** project.
2. In the BlackBerry Smartphone Simulator, click **BlackBerry Maps Demo**.
3. Open the menu and click **Invoke Location Document**.

## GPS and BlackBerry Maps Development Guide

4. Open the menu and click **View Single Location**. BlackBerry® Maps displays the following map:



5. To view more details, from the application menu, select **Details**.

## Open BlackBerry Maps to display multiple locations on a map

1. Create a `String` that contains location information for multiple locations.

```
String document = "<lbs>" + "<location lon='-8030000' lat='4326000' label='Kitchener, ON' description='Kitchener, Ontario, Canada' />" + "<location lon='-7569792'
```

### 3: Displaying location information on a BlackBerry device

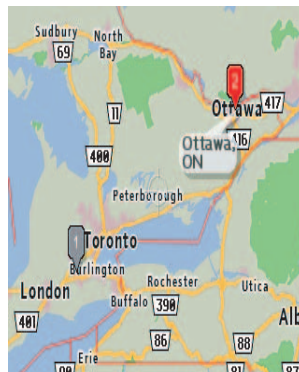
```
lat='4542349' label='Ottawa, ON' description='Ottawa, Ontario, Canada' />" + "</lbs>";
```

2. Invoke `invokeApplication()` using the `APP_TYPE_MAPS` constant parameter, a new `MapsArguments` object that uses the `ARG_LOCATION_DOCUMENT` property and the location `String`.

```
Invoke.invokeApplication(Invoke.APP_TYPE_MAPS, new MapsArguments(  
MapsArguments.ARG_LOCATION_DOCUMENT, document));
```

## View a sample application that displays multiple locations on a map

1. From the BlackBerry® Integrated Development Environment, start the **BlackBerryMapsDemo** project.
2. In the BlackBerry Smartphone Simulator, click **BlackBerry Maps Demo**.
3. Open the menu and click **Invoke Location Document**.
4. Open the menu and click **View Multiple Locations**. BlackBerry® Maps displays the following map:



## Open BlackBerry Maps to display a route between locations on a map

1. Create a `String` with route information within the `<getRoute>` elements.

```
String document = "<lbs><GetRoute>" + "<location lon='-8030000' lat='4326000'  
label='Kitchener, ON' description='Kitchener, Ontario, Canada' />" +  
"<location lon='-7569792' lat='4542349' label='Ottawa, ON' description='Ottawa,  
Ontario, Canada' />" + "</GetRoute></lbs>";
```

2. Invoke `invokeApplication()` using the `APP_TYPE_MAPS` constant parameter, a new `MapsArguments` object that uses the `ARG_LOCATION_DOCUMENT` property and the `String`.

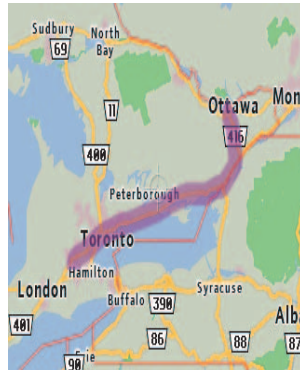
```
Invoke.invokeApplication(Invoke.APP_TYPE_MAPS, new MapsArguments(  
MapsArguments.ARG_LOCATION_DOCUMENT, document));
```

## View a sample application that displays a route between locations on a map

1. From the BlackBerry® Integrated Development Environment, start the **BlackBerryMapsDemo** project.
2. In the BlackBerry Smartphone Simulator, click **BlackBerry Maps Demo**.

## GPS and BlackBerry Maps Development Guide

3. Open the menu and click **Invoke Location Document**.
4. Open the menu and click **View Route**. BlackBerry® Maps displays the following map:



5. To view tracking information from one location to another, from the application menu, select **Directions**.

## Open BlackBerry Maps to display a custom map view

A `MapView` object represents the view that BlackBerry® Maps displays. The `MapView` object contains the latitude and longitude that BlackBerry Maps uses as the center of the map that it displays and the zoom for the map that it displays.

1. Import the `net.rim.blackberry.api.maps.MapView` class.  

```
import net.rim.blackberry.api.maps.MapView;
```
2. Create an instance of a `MapView`.  

```
MapView mapView = new MapView();
```
3. Set the latitude, longitude, and zoom values for the `MapView` object.  

```
mapView.setLatitude(4331168);  
mapView.setLongitude(-8055339);  
mapView.setZoom(10);
```
4. Invoke `invokeApplication()` using the `APP_TYPE_MAPS` constant parameter, and a new `MapsArguments` object that uses `MapView` object you created.  

```
Invoke.invokeApplication(Invoke.APP_TYPE_MAPS, new MapsArguments(mapView));
```

## View a sample application that displays a custom map view

1. From the BlackBerry® Integrated Development Environment, start the **BlackBerryMapsDemo** project.
2. In the BlackBerry Smartphone Simulator, click **BlackBerry Maps Demo**.
3. Open the menu and click **Invoke Map View**.
4. Open the menu and click **View Map**.

## Open BlackBerry Maps to display the location for an address in the contact list

You can create a BlackBerry® Application that displays a map in BlackBerry Maps for the address of a contact in the contact list. The address for the contact must be a valid address.

1. Import the `javax.microedition.pim.PIM` class and the `javax.microedition.pim.Contact` interface.

```
import javax.microedition.pim.PIM;
import javax.microedition.pim.Contact ;
```

2. Create a contact object.

```
Contact c = null;
```

3. In a try block, get the contact list.

```
try {
    ContactList contactList = (ContactList)PIM.getInstance().openPIMList(PIM.CONTACT_LIST,
    PIM.READ_WRITE);
```

4. Place the items from the contact list into an enumeration.

```
Enumeration enumContact = contactList.items();
int notFinished = 1;
```

5. Create a while loop to iterate through all the items in the enumeration.

```
while ((enumContact.hasMoreElements()) && ( notFinished == 1)) {
    c = (Contact)enumContact.nextElement();
    int[] fieldIds = c.getFieldIds();
    int id;
```

6. Search for the first contact with a valid city and region and display the address information for this contact in BlackBerry® Maps.

```
    if (c.countValues(Contact.ADDR) > 0) {
        String address[] = c.getStringArray(Contact.ADDR,0);
        if ((address[Contact.ADDR_LOCALITY] != null) && (address[Contact.ADDR_REGION]
        != null)) {
            Invoke.invokeApplication(Invoke.APP_TYPE_MAPS, new MapsArguments(c,0));
            notFinished = 0;
```

```
        }
    }
} catch (PIMException e) {
}
```

7. If a contact address is not found, display the default map in BlackBerry Maps.

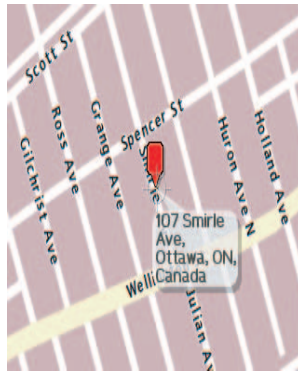
```
    if (c == null) {
        Invoke.invokeApplication(Invoke.APP_TYPE_MAPS, new MapsArguments());
    }
};
```

## View a sample application that displays the location for an address in the address book

1. On a BlackBerry® device, open the address book.

## GPS and BlackBerry Maps Development Guide

2. Add a contact to the address book. If there are no addresses in the address book, the sample will display the default BlackBerry® Maps view.
3. Open the BlackBerry® Integrated Development Environment.
4. Run the **BlackBerryMapsDemo** project.
5. In the BlackBerry Smartphone Simulator, click **BlackBerry Maps Demo**.
6. Open the menu and click **Invoke Contact**.
7. Open the menu and click **View Map**.



## Open BlackBerry Maps from the browser

A BlackBerry device user can use the browser on a BlackBerry device to access location information on a web server and display that information in BlackBerry Maps. Developers store location information in .xloc files and place the .xloc files on a web server. When a BlackBerry device user accesses the link for the .xloc file, for example <http://servername/Filename.xloc>, the browser launches BlackBerry Maps to display the location information. The browser passes the .xloc file as a String to the Location Document parser.

1. Create a file with the extension .xloc.
2. Place location information in the .xloc file. The location information is the same information you place on the right-side of the equal sign in a location document. For example, "<lbs><location lon='-7938675' lat='4367022' label='Toronto, ON' description='Toronto' zoom='10'/></lbs>";
3. On a web server, place the .xloc file in the folder where you store html files.
4. Register the following MIME type with the web server: **text/vnd.rim.location xloc**.

## Clearing data from the map that BlackBerry Maps displays

You can use the `id` and `clear` attributes of the `<lbs>` element to remove data from a map that BlackBerry® Maps displays.

## Clear data specific to a location document with an id attribute

You can only remove data from a map that is from a location document that has an `id` attribute in the `<lbs>` element.

- > Create a `String` that sets the `clear` attribute to the `id` of the location document that contains the data to remove from a map.

```
String document = "<lbs clear='Montreal' ><location lon='-7938675' lat='4367022' label='Toronto, ON' description='Toronto' zoom='10' /></lbs>";
```

## Clear data specific to all location documents with an id attribute

You can only remove data from a map that is from a location document that has an `id` attribute in the `<lbs>` element.

- > Create a `String` that sets the `clear` attribute to `Docs`.

```
String document = "<lbs clear='Docs' ><location lon='-7938675' lat='4367022' label='Toronto, ON' description='Toronto' zoom='10' /></lbs>";
```

## Clear data specific to specific location documents with an id attribute

You can only remove data from a map that is from a location document that has an `id` attribute in the `<lbs>` element.

- > Create a `String` that sets the `clear` attribute to a comma-delimited list of `ids`, for example

```
String document = "<lbs clear='MONTREAL','OTTAWA' ><location lon='-7938675' lat='4367022' label='Toronto, ON' description='Toronto' zoom='10' /></lbs>";
```

## Clear all location data from a map

You can only remove data from a map that is from a location document that has an `id` attribute in the `<lbs>` element.

- > Create a `String` that sets the `clear` attribute to `Locations`, for example

```
String document = "<lbs clear='Locations' ><location lon='-7938675' lat='4367022' label='Toronto, ON' description='Toronto' zoom='10' /></lbs>";
```

## Displaying location information in your application

A BlackBerry® Application can use the `MapField` class to display a map in a UI field. For a BlackBerry Application to use the `MapField` class, BlackBerry® Maps must be installed on the BlackBerry device and the BlackBerry device must be able to download data from a location-based services map server.

A `MapField` gets data either from a location-based services map server or the BlackBerry device with BlackBerry Maps installed. If location information is not present on the BlackBerry device, the `MapField` requests data from the location-based services map server. See the API reference for the BlackBerry Java Development Environment for more information about the `MapField` API.

## Configure a map in a UI field

Perform any of the following tasks:

Task	Steps
Set the center of a <code>MapField</code> using latitudinal and longitudinal values, in degrees.	> Invoke <code>moveTo(int latitude, int longitude)</code> . The latitude and longitude values are calculated as degrees multiplied by 100,000.
Set the center of a <code>MapField</code> to specific coordinates.	> Invoke <code>moveTo(javax.microedition.location.Coordinates coords)</code> .
Set the center of a <code>MapField</code> using pixel values.	> Invoke <code>move(int dx, int dy)</code> . The <code>move(dx, dy)</code> method moves the map in terms of relative pixels. A positive <code>dx</code> moves east if rotation is 0, and a positive <code>dy</code> moves north if rotation is 0.
Set the zoom value for a <code>MapField</code> .	> Invoke <code>setZoom(int z)</code> . Zoom values can be from 0 to 15. The zoom operation is performed to the center of the map view.
Set the rotation for a <code>MapField</code> , in degrees.	> Invoke <code>setRotation(int r)</code> . Rotation can be set from 0 to 360, clockwise. A rotation setting of 0 sets north at the top of the map view.
Retrieve the latitude of the center of a <code>MapField</code> .	> Invoke <code>getLatitude()</code> .
Retrieve the longitude of the center of a <code>MapField</code> .	> Invoke <code>getLongitude()</code> .
Retrieve the zoom value of a <code>MapField</code> .	> Invoke <code>getZoom()</code> .
Retrieve the rotation setting for a <code>MapField</code> .	> Invoke <code>getRotation()</code> .

## Converting latitudinal and longitudinal coordinates to pixel values

A BlackBerry® Application can convert latitudinal and longitudinal coordinates to field coordinates, in pixels. For latitude, north is a positive value, and south is a negative value. For longitude, east is a positive value and west is a negative value.

The `Coordinates` class represents coordinates as latitudinal, longitudinal, and altitudinal values. The latitudinal and longitudinal values are expressed in degrees using floating point values. The coordinates are given using World Geodetic System 1984 (WGS84) data.

### Convert latitudinal and longitudinal coordinates to pixels

The `XYPoint` object stores the initial coordinates and changes to the converted coordinates.

Invoke `convertWorldToField(Coordinates worldIn, XYPoint fieldOut)` or `convertWorldToField(XYPoint worldIn, XYPoint fieldOut)`.

### Convert pixels to latitudinal and longitudinal coordinates

The `XYPoint` object stores the initial coordinates and changes to the converted coordinates.

Invoke `convertFieldToWorld(XYPoint fieldIn, Coordinates worldOut)` or `convertFieldToWorld(XYPoint fieldIn, XYPoint worldOut)`.

## Setting the display size of a map in your application

A BlackBerry® Application can control the size of a `MapField` on the display screen. For example, the application can change the size of a `MapField` to make other UI components visible on the screen of a BlackBerry device.

If you set the preferred size of `MapField`, you override the dimension values passed in `layout()`. The height and width values are in pixels.

### Set the display size of a MapField

> Invoke `setPreferredSize(int preferredWidth, int preferredHeight)`.

To view complete code samples for the BlackBerry® Java Development Environment, visit [www.blackberry.com](http://www.blackberry.com).

For RIM and BlackBerry partner use only. Beta documentation. Content and software are subject to change.

## GPS and BlackBerry Maps Development Guide





©2008 Research In Motion Limited

Published in Canada.