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

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

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

The terms of use of any RIM product or service are set out in a separate license or other agreement with RIM applicable thereto. NOTHING IN THIS DOCUMENTATION IS INTENDED TO SUPERSEDE ANY EXPRESS WRITTEN AGREEMENTS OR WARRANTIES PROVIDED BY RIM FOR PORTIONS OF ANY RIM PRODUCT OR SERVICE OTHER THAN THIS DOCUMENTATION.
Chapter 3

Introduction to GPS and Wi-Fi technology on BlackBerry smartphones

Objectives

- Define GPS
- Describe the functionality of GPS on BlackBerry® smartphones
- Provide an example of how a GPS application works on a BlackBerry smartphone
- Identify the limitations and benefits of using GPS on a BlackBerry smartphone
- Define Wi-Fi® for BlackBerry smartphones
- Describe the functionality of Wi-Fi on BlackBerry smartphones
- Provide an example of how BlackBerry smartphones use Wi-Fi
- Identify the limitations and benefits of using Wi-Fi on a BlackBerry smartphone

This chapter introduces GPS and Wi-Fi technology and describes the advantages and disadvantages of implementing these features in BlackBerry smartphone applications.
GPS technology

GPS technology is made available through a series of satellites that continually transmit position information. BlackBerry smartphone applications can use this GPS information in a variety of ways, including the following:

- providing users with turn-by-turn navigation
- allowing users to share their current location with others
- geotagging photographs (automatically tagging photographs with location information)
- providing users with targeted promotions and alerts based on their current location

To use GPS information in your BlackBerry application, use the Location API for Java® ME in the javax.microedition.location package (JSR 179).

JSR 179: Location API

JSR 179 is very easy to use. The following are the three most commonly used classes:

- LocationProvider is the engine.
- Criteria is passed to LocationProvider to define the type of GPS fix.
- Location is the set of location data (such as coordinates, direction, and speed) generated by LocationProvider.

The most commonly used interfaces are as follows:

- LocationListener receives updated location data from LocationProvider at specified intervals. LocationListener also detects when GPS information is or becomes unavailable.
- ProximityListener detects proximity to a specified set of coordinates. Your application can use proximity information to present the user with appropriate location-based information or options.

Sources of GPS location information

You can use one of several modes to retrieve GPS location information. These modes are often called fixes. The time it takes to retrieve the location of the BlackBerry smartphone depends on several factors, including the retrieval mode and the GPS signal strength. In autonomous mode, typical times are less than two minutes. In assisted mode, typical times are less than 30 seconds. To retrieve and display GPS location information, the BlackBerry smartphone must support GPS technology and have a valid SIM card.

GPS location information can be unavailable if the BlackBerry device cannot receive signals from GPS satellites. This can occur when something, such as a building or dense cloud cover, obstructs the signals. When creating a GPS application for the BlackBerry smartphone, you must specify the mode (cell site, assisted, or autonomous) of location information retrieval.
**Cellsite fix**

Cellsite fixes provide location information for the cell tower the mobile device is currently using. When possible, this type of fix uses cell tower triangulation to provide more accurate information. This type of fix gets location information from the cellular network, rather than from GPS satellites.

Because cellsite fixes do not leverage the GPS satellite network, they provide the initial location information more quickly than other fix types and consume very little power. One disadvantage of a cellsite fix is that it is the least accurate of the fix types. Even when triangulation is used, cellsite fixes are typically only accurate to within 100 meters. Another disadvantage is that cellsite fixes are typically available only on CDMA, GPRS, and i-DEN networks.

**Note:**
The time required to retrieve the initial fix is often referred to as the *time to first fix*, or *TTFF*.

To function properly, both the BlackBerry smartphone and the wireless service provider must support cellsite fixes. BlackBerry smartphone users can incur some costs by using a cellsite fix.

Use the following steps to create a GPS application that uses a cellsite fix:

1. Import the following classes:
   - `javax.microedition.location.LocationProvider`
   - `javax.microedition.location.Criteria`

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

3. Invoke `criteria.setHorizontalAccuracy(NO_REQUIREMENT)` to specify that longitudinal accuracy is not required.

4. Invoke `criteria.setVerticalAccuracy(NO_REQUIREMENT)` to specify that latitudinal accuracy is not required.

5. Invoke `criteria.setCostAllowed(true)` to specify that this mode can incur cost.

6. Invoke `setPreferredPowerConsumption(POWER_USAGE_LOW)` to specify that power consumption is low.

7. Invoke `LocationProvider.getInstance()`, storing the returned object in a Location-Provider object.
   ```java
   LocationProvider provider = LocationProvider.getInstance(criteria);
   ```
### Autonomous GPS fix

With an autonomous fix, the BlackBerry smartphone receives location information directly from orbiting GPS satellites. One advantage of the autonomous fix is that it is very accurate (it is accurate to within 30 meters). Also, the autonomous fix does not require network coverage, because information comes directly from GPS satellites.

However, autonomous fixes have the highest power consumption. They also have the slowest retrieval time to first fix (up to two minutes), because the receiving device has to search for satellite signals.

Use the following steps to configure autonomous mode data retrieval:

1. Import the following classes:
   - `javax.microedition.location.LocationProvider`
   - `javax.microedition.location.Criteria`

2. Create an instance of a Criteria object.

   ```java
   Criteria criteria = new Criteria();
   ```

3. Invoke `criteria.setCostAllowed(false)` to specify that this mode does not incur cost.

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

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

### Assisted GPS

Assisted GPS fixes use both the GPS satellites and the network. In an assisted GPS fix, the receiving device communicates with the PDE server on the wireless service provider network to determine GPS satellite locations.

Because assisted GPS fix information comes directly from GPS satellites, it is very accurate (within 30 meters), and because this fix type gets satellite locations from the PDE server, it has a much shorter TTFF than autonomous GPS.

Assisted GPS fixes have higher power consumption than cellsite fixes, but lower power consumption than autonomous fixes. Some disadvantages of assisted GPS fixes is that they require network coverage, and users can incur some costs by using this type of fix. Additionally, to function properly, both the BlackBerry smartphone and the wireless service provider must support assisted GPS fixes.

Use the following steps to create a GPS application that uses assisted GPS data retrieval:

1. Import the following classes:
• javax.microedition.location.LocationProvider
• javax.microedition.location.Criteria

2. Create an instance of a Criteria object.

   Criteria criteria = new Criteria();

3. Invoke criteria.setHorizontalAccuracy(NO_REQUIREMENT) to specify that longitudi-
   nal accuracy is not required.

4. Invoke criteria.setVerticalAccuracy(NO_REQUIREMENT) to specify that latitudi-
   nal accuracy is not required.

5. Invoke criteria.setCostAllowed(true) to specify that this mode can incur cost.

6. Invoke criteria.setPreferredPowerConsumption(int level) using either
   POWER_USAGE_MEDIUM, or NO_REQUIREMENT as the parameter to specify power con-
   sumption is medium or not required.

7. Invoke LocationProvider.getInstance(), storing the returned object in a Location-
   Provider object.

   LocationProvider provider = LocationProvider.getInstance(criteria);

---

**Bluetooth-enabled GPS**

If there is no built-in GPS receiver on the BlackBerry smartphone, users can still receive GPS informa-

---

**BlackBerry Maps location information**

You can create a BlackBerry smartphone application that interacts with BlackBerry Maps. BlackBerry Maps is a map and location client application that can display a map for a location, the location of the BlackBerry smartphone, a route from a starting location to a specific ending location, and points of interest on a map. BlackBerry Maps is available on BlackBerry smartphones that run BlackBerry Device Software version 4.2 and later.

There are several ways to integrate BlackBerry Maps information with your application.
Open BlackBerry Maps from your application

You can invoke BlackBerry Maps from your application. Place location information into a `String` variable called a `location document`. You then invoke `net.rim.blackberry.api.invoke.Invoke.invokeApplication()` using as parameters the `net.rim.blackberry.api.invoke.Invoke.APP_TYPE_MAPs` constant and a new instance of a `net.rim.blackberry.api.invoke.MapsArguments` object created using the `net.rim.blackberry.api.invoke.MapsArguments.ARG_LOCATION_DOCUMENT` constant and the `String` variable with the location information. BlackBerry Maps starts and displays the information in the location document.

Open BlackBerry Maps from the BlackBerry Browser

You can let BlackBerry smartphone users use the BlackBerry® Browser to access location information on a web server and display that information in BlackBerry Maps. Create a file with the extension `.xloc` and place location information in the file. The location information is the same information that you assign to the location document to start BlackBerry Maps from a BlackBerry application. Next, place the `.xloc` file on a web server. When a BlackBerry smartphone user accesses the link for the `.xloc` file, the browser passes the `.xloc` file as a `String` to the location document parser. BlackBerry Maps starts and displays the information in the `.xloc` file.

Open your application from BlackBerry Maps

To allow users to access your application from BlackBerry Maps, `register a menu item` with BlackBerry Maps, then create and populate the menu item.
Introduction to GPS and Wi-Fi technology on BlackBerry smartphones

Quiz

1. Which of the following do you use to include GPS information in your application?
   A. Java API for BlackBerry
   B. Location API for Java ME
   C. Java for GPS objects
   D. Any BlackBerry API

2. Which class do you use to specify the type of GPS fix?
   A. Criteria
   B. Data
   C. Speed and coordinates
   D. Both A and B

3. Which of the following statements is accurate?
   A. The JSR 179 LocationListener interface receives data from LocationProvider.
   B. The JSR 179 LocationListener interface detects proximity to a specified set of coordinates.
   C. The JSR 179 LocationListener interface generates location data such as coordinates, direction, and speed.
   D. The JSR 179 LocationListener interface is not part of the JSR 179.

4. Compared with autonomous mode, assisted mode is:
   A. Faster
   B. Slower
   C. Neither faster or slower
   D. Faster or slower, depending on conditions
5. When creating a GPS application, which of the following can you specify?
   A. Your current location
   B. The fix type
   C. The BlackBerry smartphone model
   D. The GPS satellite you intend to use

6. Assisted fixes ______ incur network charges.
   A. Never
   B. Sometimes
   C. Always

7. From which of the following do autonomous GPS fixes receive information?
   A. The cellular system
   B. Orbiting GPS satellites
   C. The cellphone itself
   D. A PDE Server on the carrier network

8. How accurate are autonomous GPS fixes?
   A. Within 3000 meters
   B. Within 300 meters
   C. Within 30 meters
   D. Within 3 meters

9. Compared to assisted and cellsite fixes, autonomous fixes use ______
   A. The least power
   B. The same power
   C. The most power
   D. More power than cellsite fixes but less than assisted fixes
10. Compared to assisted and cellsite fixes, autonomous fixes take ______ time for the initial fix
   A. The longest
   B. The shortest
   C. An intermediate
   D. No

11. From which of the following sources do assisted GPS fixes receive information? Choose all that apply.
   A. The cellular system
   B. Orbiting GPS satellites
   C. The mobile device
   D. A PDE server on the carrier network

12. On which of the following are BlackBerry Maps included?
   A. All BlackBerry smartphones
   B. BlackBerry® Curve™ Series only
   C. BlackBerry Device Software version 3.0 and later
   D. BlackBerry Device Software version 4.2 and later

13. To invoke BlackBerry Maps from your application, which of the following do you pass as parameters to the invokeApplication method? Choose all that apply.
   A. Location information as a string
   B. The APP_TYPE_MAPS constant
   C. A new instance of the MapsArgument class
   D. An instance of the BlackBerry Maps application

14. Which of the following do you create to let users access position data on the server?
   A. An .xloc file containing a serialized Location object
   B. A .txt file with position information as a string
C. A .bbmap file containing latitude and longitude data

D. An .xloc file containing position information as a string
Introduction to GPS and Wi-Fi technology on BlackBerry smartphones

Answers

1. B
2. A
3. A
4. A
5. B
6. B
7. B
8. C
9. C
10. A
11. B, D
12. D
13. A, B, C
14. D
Wi-Fi

In a traditional mobile network, a service provider uses a series of cellular base stations and cell towers to provide wireless access (for a fee) to customers for phone, Internet, and email service. Cellular networks are proprietary and use technologies such as CDMA, GSM, and 3G. Cellular networks typically cover huge areas, but are susceptible to interference from buildings and natural geographical barriers (such as trees and hills).

Conversely, you can set up a Wi-Fi network simply by plugging a wireless router (also called a WLAN access point) into any broadband modem with Internet access. Mobile devices with a Wi-Fi receiver can then use this Wi-Fi network (called a hotspot) instead of using the cellular network. Hotspots have a very limited range (up to 50 meters), but are very fast and are less susceptible to interference.

Along with the limited coverage, Wi-Fi has the disadvantage of being less secure than cellular networks. Many businesses, such as cafes, hotels, and campgrounds, set up hotspots to attract customers. In many cases, they leave the network unsecured, so anyone with a Wi-Fi receiver can access the network.

Use Wi-Fi in your application

You can configure your application to use Wi-Fi, as well as to determine the presence and status of a Wi-Fi receiver on the BlackBerry smartphone.

Determine whether the BlackBerry smartphone supports Wi-Fi

Not all BlackBerry smartphones are Wi-Fi-capable. To determine if the BlackBerry smartphone can use Wi-Fi, invoke RadioInfo.areWAFsSupported(int wafs).

Determine if the Wi-Fi transceiver is turned on

Use the following steps to determine if the Wi-Fi transceiver is turned on:

1. Import the net.rim.device.api.system.RadioInfo class.

2. Create an IF statement that tests the value of RadioInfo.WAF_WLAN and the value returned by RadioInfo.getActiveWAFs().

   if ( ( RadioInfo.getActiveWAFs() & RadioInfo.WAF_WLAN ) != 0 ) {
   ... }

Turn the Wi-Fi transceiver on

To have your application turn the Wi-Fi transceiver on, invoke Radio.activateWAFs(int WAFs).
Determine if the Wi-Fi transceiver is connected to a WLAN access point

Use the following steps to determine if the Wi-Fi transceiver is connected to a wireless access point:

1. Import the `net.rim.device.api.system.WLANInfo` class.

2. Create an IF statement that tests the value of `WLANInfo.WLAN_STATE_CONNECTED` and the value returned by `WLANInfo.getWLANState()`. The `WLANInfo.getWLANState()` method checks if a BlackBerry smartphone has an IP address and can transfer data over a Wi-Fi network. If the transceiver for the WLAN wireless access family is off, this method returns `WLANInfo.WLAN_STATE_DISCONNECTED`.

   ```java
   if (WLANInfo.getWLANState() == WLANInfo.WLAN_STATE_CONNECTED) {...}
   ```

Open a Wi-Fi connection

BlackBerry smartphones that use BlackBerry® Mobile Data System or BlackBerry® Internet Service for connectivity switch automatically, and with no additional fee or configuration, to Wi-Fi if it is available. This automatic switching is called `least cost routing`. If you want your application to use Wi-Fi for other connection types (such as WAP or TCP), use the following steps to explicitly specify Wi-Fi as the connection type:

1. Import the following class
   - `javax.microedition.io.Connector`

2. Import the `javax.microedition.io.HttpConnection` interface.

3. Invoke `Connector.open()`, specify HTTP as the protocol, and append the `interface=wifi` parameter to the end of the web address string value.

4. Cast the returned object as an `HttpConnection` or a `StreamConnection` object.

   ```java
   HttpConnection conn = null;
   String URL = "http://www.myServer.com/myContent;device-side=true;interface=wifi";
   con = (HttpConnection) Connector.open(URL);
   ```

Retrieve the Wi-Fi connection status and profile

Your BlackBerry smartphone application can retrieve status information about the Wi-Fi connection, such as the data rate of the connection, the SSID of the WLAN access point, and the name of the `Wi-Fi profile` in use. To support this feature, the BlackBerry smartphone must be using a Wi-Fi network.

1. Import the `net.rim.device.api.system.WLANInfo` class.
2. Invoke `WLANInfo.getAPInfo()`, storing a reference to `WLANInfo.WLANAPInfo` that this method returns. The `WLANInfo.WLANAPInfo` object contains a snapshot of the current wireless network.

```java
WLANInfo.WLANAPInfo info = WLANInfo.getAPInfo();
```

If the BlackBerry device is not connected to an access point, the `WLANInfo.getAPInfo()` method returns null.
1. Compared to mobile networks, Wi-Fi networks are (choose all that apply):
   A. Very fast
   B. More susceptible to interference
   C. Less susceptible to interference
   D. More limited in range

2. Which of the following can you do with your BlackBerry smartphone application? Choose all that apply.
   A. Ensure a BlackBerry smartphone Wi-Fi transciever is turned on
   B. Retrieve details about the Wi-Fi connection
   C. Expand the hotspot to allow access to the BlackBerry smartphone

3. Which of the following conditions are required for a BlackBerry smartphone to use least cost routing?
   A. The BlackBerry smartphone must be in range of a hotspot
   B. The BlackBerry smartphone must use either BlackBerry MDS or BlackBerry Internet Service
   C. The BlackBerry smartphone must be out of range of a mobile network
   D. The application must include the `net.rim.device.api.system.WLANInfo` class.

4. Which of the following do you do to explicitly specify Wi-Fi as the connection type for your application?
   A. Append the `interface=wifi` parameter to the web address
   B. Use the HTTPS protocol
   C. Cast the returned object as a `WiFiConnection` object
   D. Append the `wifi=interface` parameter to the web address
5. To determine if the Wi-Fi transceiver is turned on, you must:

A. **Determine whether the** `RadioInfo.WAF_WLAN` **method returns the value**
   `getActiveWAF`

B. **Determine whether the** `getActiveWAF` **method returns the value**
   `RadioInfo.WAF_WLAN`

C. **Determine whether the BlackBerry smartphone is outside of a hotspot**

D. **Invoke the** `WLANInfo.getAPIInfo` **method**
Answers

1. A, C, D
2. A, B
3. A, B
4. A
5. B
GPS technology provides users with location information. BlackBerry smartphones get GPS information using one of three fix types.

- **Cellsite fixes** use the mobile network to provide the location of the current cell tower. This fix type is faster but less accurate than the other types, and can incur a network usage cost.

- **Autonomous fixes** get GPS data directly from GPS satellites. This fix type is very accurate (to within 30 meters) and does not incur network charges, but it has the slowest time to first fix.

- **Assisted fixes** get GPS data directly from GPS satellites, but use the PDE server on the mobile network to locate the GPS satellites for the first fix. This fix type is very accurate (to within 30 meters) and has a fast time to first fix, but can incur a network charge.

BlackBerry smartphones without a GPS receiver can use any of these fix types by communicating with a Bluetooth-enabled GPS receiver (this capability is available in BlackBerry Device Software version 4.2 and later). Communication with a Bluetooth-enable GPS receiver requires no additional configuration within your application. You can also incorporate BlackBerry Maps functionality in your application to provide maps, routes, and directions to points of interest.

Some BlackBerry smartphones support Wi-Fi connections. If the BlackBerry smartphone uses BlackBerry MDS or BlackBerry Internet Service to connect, it uses least cost routing, meaning it switches automatically to Wi-Fi when the user enters a hotspot. You can configure your application to use Wi-Fi, detect the Wi-Fi capabilities of the BlackBerry smartphone, and retrieve status information about the Wi-Fi connection.
Review Questions

1. Briefly describe the changes you must make to your application to enable communication with a Bluetooth-enabled GPS receiver.

2. Briefly describe the three GPS fix types in terms of the location information provided and how that information is relayed to the mobile device.

3. Compare and contrast the three GPS fix types in terms of accuracy, power consumption, availability, and network cost.

4. Describe the sequence of steps required in code to access an autonomous fix.

5. Describe the purpose of each of the following classes in the JSR 179 Location API:
   - LocationProvider
   - Criteria
   - LocationListener
   - ProximityListener

6. Compare and contrast the functionality of Wi-Fi and mobile networks in terms of coverage, interference, security, and speed.

7. To open a Wi-Fi connection, your application must determine if the Wi-Fi transceiver is turned on, then open a connection. Describe the steps required to do this, in terms of the code used.