Advanced Java Application Development for the BlackBerry Smartphone

BlackBerry Academic Program
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 5
Introduction to multimedia application development for the BlackBerry smartphone

Objectives

- Explain the importance of playing media within a media application
- Create a media application
- Play media in a BlackBerry smartphone application
- Discuss the use of media events
- Send and receive media events
- Discuss audio routing
- Create an application that records audio
- Identify supported audio and video formats
- Identify optimal settings for audio and video on BlackBerry® smartphones

This chapter outlines the multimedia capabilities that you can add to your BlackBerry smartphone applications. Use the information in this chapter to create a BlackBerry smartphone application that plays media in the BlackBerry® Browser or in the Media application on a BlackBerry smartphone. You can also create a BlackBerry smartphone media application that can play audio, video, and binary SVG content, record audio and video, or send audio to a Bluetooth enabled headset.
Playing media

There are several options you can choose from to play media in your BlackBerry smartphone applications.

- You can create a media application that can play a variety of media types and formats.
- You can play media using the BlackBerry Browser.
- You can play media using the BlackBerry smartphone media application (on BlackBerry smartphones that support this option).

Creating a media application

You can create a media application by using the API items in the javax.microedition.media (JSR 135) package. You can choose what functionality to add to your media application.

Create an application that plays a sequence of tones

To play a sequence of tones, implement ToneControl to permit playback of a BlackBerry smartphone user-defined sequence of tones in an unvarying pitch.

Tempo is a measurement of the bpm with 1 beat equal to 1/4 note. Determine the tempo by multiplying the tempo modifier by four; the tempo modifier must fall within the byte range of 1 to 127. Tempos in the range of 20 bpm to 508 bpm equate to a tempo modifier range of 5 to 127.

Create an application that plays media from a web address

To create an application that plays media from a web address, invoke Manager.createPlayer(String locator). The String parameter is a web address that describes the media content.

Create an application that plays media from an input stream

To create an application that plays media from an input stream, perform the following steps:

1. Invoke Manager.createPlayer(InputStream stream, String type). The type parameter represents the type of input media content.

   Check for a MediaException if the content type is not specified.

   ```
   RecordStore recSt; 
   int recId;
   ```
try {
    InputStream inpStr = new ByteArrayInputStream((store.getRecord(recId));
    Player p = Manager.createPlayer(inpStr, "audio/mpeg");
    p.start();
}

Create an application that plays streaming media

BlackBerry smartphones that have BlackBerry Device Software version 4.3.0 or later installed can support RTSP functionality.

1. Invoke Manager.createPlayer(String locator). The String parameter represents an RTSP locator.

2. Manager.createPlayer("rtsp://streaming.rim.com/streaming_video.3gp");

Create an application that displays a video in a field.

1. Create the Player, VideoControl, and Field variables.
   Player _videoPlayer;
   VideoControl _videoControl;
   Field videoField;

2. Start a try block.
   try {

3. Invoke Manager.createPlayer(String locator). The String parameter is an address with URI syntax that describes the video content. Store a reference to the Player object that the call to createPlayer(String locator) returns.
   _videoPlayer = Manager.createPlayer("file:///SDCard/BlackBerry/videos/soccer1.avi");

4. To permit the application to get the information that it requires to get media resources, invoke Player.realize().
   _videoPlayer.realize();

5. Invoke Player.getControl(). The String parameter is a parameter that represents the VideoControl class. Cast the returned object as a VideoControl object.
   _videoControl = (VideoControl)_videoPlayer.get-Control("javax.microedition.media.control.Video-Control");
6. To initialize the mode that a videoField uses to display the video, invoke VideoControl.initDisplayMode(int mode, Object arg). The String parameter is an arg value that specifies the UI primitive that displays the video.

For example, in a BlackBerry Application, use net.rim.device.api.ui.Field as the arg parameter, casting the object that this method returns as a Field object. For more information about initializing a videoField, see the API reference for the BlackBerry Java Development Environment.

```java
videoField = (Field) _videoControl.initDisplayMode(
    VideoControl.USE_GUI_PRIMITIVE, "net.rim.device.api.ui.Field" );
```

7. Check for exceptions within the try block.

```java
} catch ( Exception e ) {
    System.out.println( "Exception: " + e.toString() )
};
```

Create an application that displays an image from the camera viewFinder.

1. Invoke Manager.createPlayer(String locator). The String parameter is an address with URI syntax that describes the image content.

   ```java
   Player cameraPlayer = Manager.createPlayer( "capture://
       video?encoding=jpeg" );
   ```

2. To permit the application to get the information that it requires to get media resources, invoke Player.realize().

   ```java
   cameraPlayer.realize();
   ```

3. Invoke Player.getControl(). The String parameter is a value that represents the VideoControl class. Cast the returned object as a VideoControl object.

   ```java
   VideoControl videoControl = (VideoControl)cameraPlayer.getControl( "javax.microedition.media.control.VideoControl" );
   ```

4. To initialize the mode that a videoField uses to display the video, invoke VideoControl.initDisplayMode(int mode, Object arg).

   ```java
   Field videoField = (Field) _videoControl.initDisplayMode(
       VideoControl.USE_GUI_PRIMITIVE, "net.rim.device.api.ui.Field" );
   ```

To capture the image from the camera view, invoke VideoControl.getSnapshot().

```java
int[] imageByte = videoControl.getSnapshot(null);
```

**Code sample: playing a sequence of tones**

```java
// "Mary Had A Little Lamb" has "ABAC" structure

// Use block to repeat "A" section
```
byte tempo = 30; // 30 x 4 = tempo of 120 bpm
byte duration = 8; // Note length 8 (quaver) = 1/8th of a note duration
byte C4 = ToneControl.C4; // C note value = 60 (middle C)
byte D4 = (byte)(C4 + 2); // D note value = 62 (a whole step)
byte E4 = (byte)(C4 + 4); // E note value = 64 (a major third)
byte G4 = (byte)(C4 + 7); // G note value = 67 (a fifth)
byte rest = ToneControl.SILENCE; // rest
byte[] mySequence = {
    ToneControl.VERSION, 1, // version 1
    ToneControl.TEMPO, tempo, // configure tempo
    
    //
    // Start define "A" section
    ToneControl.BLOCK_START, 0,
    
    //
    // Content of "A" section
    E4, duration, D4, duration, C4, duration, E4, duration,
    E4, duration, E4, duration, E4, duration, rest, duration,
    
    //
    // End define "A" section
    ToneControl.BLOCK_END, 0,
    
    //
    // Play "A" section
    ToneControl.PLAY_BLOCK, 0,
    
    //
    // Play "B" section
    D4, duration, D4, duration, D4, duration, rest, duration,
E4, duration, G4, duration, G4, duration, rest, duration,
   //
   // Repeat "A" section
ToneControl.PLAY_BLOCK, 0,
   //
   // Play "C" section
D4, duration, D4, duration, E4, duration, D4, duration, C4, duration
};

try{
    Player p = Manager.createPlayer(Manager.TONE_DEVICE_LOCATOR);
p.realize();
    ToneControl c = (ToneControl)p.getControl("ToneControl");
c.setSequence(mySequence);
p.start();
} catch (IOException ioe) {
} catch (MediaException me) {

**Code sample: playing media from an input stream**

//Determine the supported content types
String types[] = Manager.getSupportedContentTypes(null);
for (int cnt = types.length - 1; cnt >= 0; --cnt) {
    if (types[cnt].equals("audio/mpeg")) {
        try {
            //Download the MP3 file
            Class clazz = Class.forName("com.rim.samples.AudioDemo");
            InputStream is = clazz.getResourceAsStream("/jungleYell.mp3");
            //Create an instance of the player from the InputStream
Player player = javax.microedition.media.Manager.createPlayer(is, "audio/mpeg");
player.realize();
player.prefetch();
//start the player
player.start();
} catch (Exception ex) { }
}

else if (types[cnt].equals("audio/x-wav ")) {
  //this is where you play wav files
}
else if (types[cnt].equals("audio/midi ")) {
  //this is where you play midi files
}


Playing media using the BlackBerry Browser

Your BlackBerry application can play media in the BlackBerry Browser by using the methods of the Browser class and BrowserSession class.

Play audio in the BlackBerry Browser

1. Import the required classes.
import net.rim.blackberry.api.browser.Browser;
import net.rim.blackberry.api.browser.BrowserSession;

2. Invoke Browser.getDefaultSession().
BrowserSession soundclip = Browser.getDefaultSession();
3. **Invoke** `BrowserSession.displaypage()`.
   
   ```java
   soundclip.displayPage("file:///SDCard/BlackBerry/music/yourFile.mp3");
   ```

---

### Play video in the BlackBerry Browser

1. **Import the required classes.**
   ```java
   import net.rim.blackberry.api.browser.Browser;
   import net.rim.blackberry.api.browser.BrowserSession;
   ```

2. **Invoke** `Browser.getDefaultSession()`.
   ```java
   BrowserSession videoclip = Browser.getDefaultSession();
   ```

3. **Invoke** `BrowserSession.displaypage()`.
   ```java
   videoclip.displayPage("file:///SDCard/BlackBerry/Video/soccer-game.avi");
   ```

---

### Playing media within a media application

On phones that support the BlackBerry smartphone media application, you can use the media application to play media that is stored on a local device, media that is streamed from a remote device, or both local and streamed media.

---

### Accessing stored media

You can use the Connector class and FileConnection interface to access media stored on the BlackBerry smartphone or a microSD card.

---

#### Access media on a BlackBerry smartphone

1. **Invoke** `Connector.open(file)`. Specify the file protocol and the location of the media file on the BlackBerry smartphone.
   ```java
   FileConnection fconn = (FileConnection)Connector.open("file:///store/home/user/audio/newfile.mp3");
   ```

#### Access media on a microSD card

1. **Invoke** `Connector.open(file)`. Specify the file protocol and the location of the media file on the microSD card.
   ```java
   FileConnection fconn = (FileConnection)Connector.open("file:///SDCard/music/newfile.mp3");
   ```
On BlackBerry smartphones that include the BlackBerry smartphone media application, a BlackBerry smartphone application can use the javax.microedition.content and net.rim.device.api.content packages to start the BlackBerry smartphone media application with or without passing media content to load.

## Playing media in a BlackBerry smartphone application

You can create a BlackBerry Application that uses the Player interface and the javax.microedition.media (JSR 135) package to play media. The Player interface provides methods to manage the different states of a media application and control the playback of media files. An object of a class that implements the Player interface can be in one of five states.

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNREALIZED</td>
<td>An instance of a class that implements the Player interface is constructed.</td>
</tr>
<tr>
<td>REALIZED</td>
<td>An instance of a class that implements the Player interface can locate and get resources on a server or file system.</td>
</tr>
<tr>
<td>PREFETCHED</td>
<td>An instance of a class that implements the Player interface can perform tasks that are required to play media.</td>
</tr>
<tr>
<td>STARTED</td>
<td>An instance of a class that implements the Player interface can start playing media.</td>
</tr>
<tr>
<td>CLOSED</td>
<td>An instance of a class that implements the Player interface can release the resources it is using.</td>
</tr>
</tbody>
</table>
Chapter 5

Change the state of the Player

<table>
<thead>
<tr>
<th>Task</th>
<th>Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change the state of a player from UNREALIZED to REALIZED.</td>
<td>Invoke Player.realize().</td>
</tr>
<tr>
<td>Change the state of a player from REALIZED to PREFETCHED.</td>
<td>Invoke Player.prefetch().</td>
</tr>
<tr>
<td>Change the state of a player from PREFETCHED to STARTED.</td>
<td>Invoke Player.start().</td>
</tr>
<tr>
<td>Change the state of a player from STARTED to CLOSED.</td>
<td>Invoke Player.close().</td>
</tr>
</tbody>
</table>

Play media in a BlackBerry Application

Use the procedures in this section to play media in a BlackBerry Application.

Start the media application, and determine what controls it supports

1. Prepare the media application by performing the following actions:
   * Invoke Player.realize().
   * Invoke Player.prefetch().

2. Start the media application by invoking Player.start(). An instance of a class that implements the Player interface returns to the PREFETCHED state when you invoke Player.stop() or when it reaches the end of the media file.
   ```java
   try {
     Player p = Manager.createPlayer("www.test.rim.net/abc.wav");
     p.start();
   } catch (MediaException pe) {
   } catch (IOException ioe) {
   }
   
   3. Invoke Player.getControls().
      To provide additional functionality for a media application, use one or more of the controls that the media application supports.
      You can use one object to access multiple controls. For example, one object can be both a VolumeControl and a ToneControl. The javax.microedition.media package contains a number of Control interfaces. For more information about the javax.microedition.media package, see the API Reference.
Provide video playback support

1. To download a VideoControl object, invoke Player.getControls().

2. Open a file connection to a video file.
   FileConnection con = (FileConnection) Connector.open("file:///SDCard/BlackBerry/videos/myvid.mp4", Connector.READ);

3. Open and return an input stream for a connection.
   InputStream is = con.openInputStream();

4. Initialize and prepare the application.
   Player player = Manager.createPlayer(is, "video/mp4");
   player.realize();

5. Get the video control.
   VideoControl videoControl = (VideoControl) player.getControl("VideoControl");
   videoControl.initDisplayMode(VideoControl.USE_DIRECT_VIDEO, this);

6. To embed the video control in the canvas, specify the properties of the video control.
   videoControl.setDisplayLocation(20, 30);
   videoControl.setDisplaySize(160, 90);
   videoControl.setVisible(true);

7. Implement the methods of the VideoControl interface to give a BlackBerry Java Application a variety of video playback support features. Perform one of the following actions:
   • control the mode of video display through the use of USE_GUI_PRIMITIVE or USE_DIRECT_VIDEO
   • control the location of the video with respect to the canvas that displays the video
   • access the x-coordinate and the y-coordinate of the video with respect to the GUI object that displays the video
   • display or hide video
   • resize the video image

Adjust the volume of the media application

1. Specify the volume of the media application.
   VolumeControl volume = (VolumeControl) player.getControl("VolumeControl");
   volume.setLevel(80);

2. Perform one of the following actions:
Capture volume key events in an MIDP application.
Canvas.keyPressed():
protected void keyPressed(int keycode) {
    if(keycode == -150) { //volume up
} else if(keycode == -151) {} //volume down
}

Capture volume key events in a RIM® UI application.
Screen.keyControl():
protected boolean keyControl(char c, int status, int time) {
    if(c == Characters.CONTROL_VOLUME_UP) {
    } else if(c == Characters.CONTROL_VOLUME_DOWN) {}
}

Record changes to the volume key

1. Override Screen.keyControl().
   protected boolean keyControl(char c, int status, int time) {

2. Identify the volume key that the BlackBerry user changed.
   if(c == Characters.CONTROL_VOLUME_UP) {
   //perform tasks
} else if(c == Characters.CONTROL_VOLUME_DOWN) {
   //perform tasks
}

Close the media application

1. Invoke Player.stop().
2. Invoke Player.close().

Code sample: playing media from a web address

The following sample playes media from a web address, formatted as follows: rtsp://streaming.rim.com/streaming_video.3gp.

private void initVideo(String url) {
    try {
        _player = javax.microedition.media.Manager.createPlayer(url);
    }
}
_player.realize();
_vc = (VideoControl) _player.getControl("VideoControl");
if (_vc != null)
{
    _videoField = (Field) _vc.initDisplayMode( VideoControl.USE_GUI_PRIMITIVE, "net.rim.device.api.ui.Field");
    _vc.setVisible(true);
}
} catch(MediaException pe) {
    System.out.println(pe.toString());
} catch (IOException ioe) {
    System.out.println(ioe.toString());
}
1. You can start the BlackBerry smartphone media application in which of the following states:
   A. with content already loaded
   B. without content loaded
   C. Either A or B

2. To stream data from a remote source to the media application, you must perform which of the following activities? Choose all that apply.
   A. format a storage device to accept buffered data
   B. allocate memory for decompression
   C. buffer the source
   D. specify how the media application reads the data

3. Your BlackBerry Application can play media in the BlackBerry Browser by using the methods of which of the following classes? Choose all that apply.
   A. the AudioVideo class
   B. the Browser class and AudioVideo class
   C. the Browser class and BrowserSession
   D. the BrowserSession class

4. To play audio on a BlackBerry smartphone, you can use the API items in which package? Choose all that apply.
   A. the javax.superplayer.media (JSR 135) package
   B. the javax.microedition.media (JSR 135) package
   C. the javax.macroedition.media (JSR 82) package
   D. the javax.microedition.media (JSR 82) package

5. You can create a media application to perform which of the following actions?
   A. play media from a web address
   B. display an image from the camera viewFinder
C. play media from an input stream
D. any of the above
E. none of the above

6. You can use the Player interface to perform which of the following actions? Choose all that apply.
   A. manage the different states of a media application
   B. navigate through web pages
   C. allocate memory for buffering streams
   D. control the playback of media files

7. An object of a class that implements the Player interface can be in which of the following states? (Circle the five that apply.)
   A. REALIZED
   B. UNREALIZED
   C. FORMATTED
   D. UNFORMATTED
   E. FETCHED
   F. PREFETCHED
   G. UNFETCHED
   H. CLOSED
   I. OPENED
   J. STARTED
   K. STOPPED

8. You can create a BlackBerry Application that uses the Connector class and FileConnection interface to access media stored on which of the following devices? Choose all that apply.
   A. a microSD card
   B. USB drive
   C. BlackBerry smartphone
Answers

1. C
2. C and D
3. C
4. B
5. D
6. A and C
7. A, B, F, H, J
8. A and C
Receiving and sending media events

To create a BlackBerry Java application that listens for and sends media application events, you can use the API items in the javax.microedition.media (JSR 135) package.

### Listen for changes to the state of the media application.

1. **Implement** `PlayerListener`.
2. To register the `PlayerListener`, **invoke** `addPlayerListener`.
   ```java
   private void doPlay()
   throws IOException, MediaException
   {Player p = Manager.createPlayer("www.rim.com/rim.mp3");
   p.addPlayerListener(this);
   p.realize();
   p.prefetch();
   p.start();
   }
   ```

### Send the media application event to the registered `PlayerListener`.

1. **Invoke** `playerUpdate(Player player, String event, Object eventData)`.
   ```java
   { }
   ```
2. Release resources when the end of the media event is received
   ```java
   if ( event == PlayerListener.END_OF_MEDIA ){
   }
   ```
3. Add code for actions if the end of media is reached.
   ```java
   player.close();
   }
   ```
Routing audio

You can use the AudioPathControl API to specify the route that your application uses to play back and record audio. For example, you can specify that audio is played or recorded using the speaker and microphone built into the BlackBerry smartphone, using a Bluetooth technology headset, or using a wired headset.

Supported audio paths

- Bluetooth—AUDIO_PATH_BLUETOOTH—Supports a Bluetooth SCO link device, such as a car kit.
- A2DP—AUDIO_PATH_BLUETOOTH_A2DP—Supports a Bluetooth A2DP profile devices, such as a stereo Bluetooth headset.
- Handset—AUDIO_PATH_HANDSET—Supports the earpiece on the BlackBerry smartphone.
- Handsfree—AUDIO_PATH_HANDSFREE—Supports the speaker on the BlackBerry smartphone.
- Headset—AUDIO_PATH_HEADSET—Supports a wired headset that is connected through the headset jack.
- Headset Handsfree—AUDIO_PATH_HEADSET_HANDSFREE—Supports the speaker on the BlackBerry smartphone and a wired headset that is connected through the headset jack.

Audio path sample code

```java
try {
    Player p =
    javax.microedition.media.Manager.createPlayer("http://mycompany/test.mp3");
    p.realize();
    p.prefetch();
    Control[] c = p.getControls();
    for(int i=c.length-1; i>=0; --i) { //iterate through player controls
        if(c[i] instanceof AudioPathControl) { //Retrieve Audio Path Control
            AudioPathControl apc = (AudioPathControl)c[i];
            try{
```

// set Audio Path -- this example routes the audio to a Bluetooth device.
apc.setAudioPath(AudioPathControl.AUDIO_PATH_BLUETOOTH);

// get Audio Path
System.out.println(apc.getAudioPath());

} catch(Exception e) {
System.err.println(e.toString());
}

} catch(InterruptedException e) {
System.err.println(e.toString());

} catch(Exception e) {
System.err.println(e.toString());

}

p.start();

}
1. Which of the following statements is correct?

   A. You can use the AudioPathControl API to specify the route that your application uses to play back audio. You can record audio only using the built in microphone.

   B. You can use the AudioPathControl API to specify the route that your application uses to play back and record audio.

   C. You cannot configure the route that your application uses to play back and record audio. Only the BlackBerry smartphone user can control audio routing.
Answers

1. B
Recording audio

The BlackBerry smartphone uses two formats to record audio: AMR (default) and 8 kHz mono-16-bit PCM.

1. In a class that extends Thread, create a variable of type Player, a variable of type RecordControl for recording media from media application, a variable of type ByteArrayOutputStream for the audio stream, and a byte array variable to store the OutputStream data.

   ```java
   final class VoiceNotesRecorderThread extends Thread
   {
      private Player _player;
      private RecordControl _rcontrol;
      private ByteArrayOutputStream _output;
      private byte _data[];
   }
   ```

2. Create a constructor for the class.

   ```java
   VoiceNotesRecorderThread() {}
   ```

3. Create a method that returns the size of the output stream buffer.

   ```java
   private int getSize()
   {
      return (_output != null ? _output.size() : 0); }
   ```

4. Create a method that returns an array that contains the contents of the output stream buffer.

   ```java
   private byte[] getVoiceNote()
   {
      return _data; }
   ```

5. In a try block in your implementation of the run method, create an application that records audio. Specify the encoding to use as a parameter to the Manager.createPlayer(String locator) method. You can use the following supported locator strings:

   ```java
   public void run() {
      try {
         _player = Manager.createPlayer("capture://audio");
   ```

<table>
<thead>
<tr>
<th>Format</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMR</td>
<td>capture://audio</td>
</tr>
<tr>
<td>AMR</td>
<td>capture://audio?encoding=amr or capture:// audio?encoding=audio/amr</td>
</tr>
<tr>
<td>PCM</td>
<td>capture://audio?encoding=pcm or capture:// audio?encoding=audio/basic</td>
</tr>
</tbody>
</table>

6. Change the Player to the REALIZED state.

   ```java
   _player.realize();
   ```
7. Get a control to record media from an application.
   \_rcontrol = (RecordControl)\_player.getControl("RecordControl");

8. Create a ByteArrayOutputStream to record the audio stream.
   \_output = new ByteArrayOutputStream();

9. Configure the output stream to which the application records data.
   \_rcontrol.setRecordStream(_output);

10. Start recording the audio and start playing the media from the Player.
    \_rcontrol.startRecord();
    \_player.start();

11. In a catch block, specify that a notification dialog displays with a message about an exception, if one occurs.
    } catch (final Exception e){
        UiApplication.getUiApplication().invokeAndWait(new Runnable()
        {
            public void run() {
                Dialog.inform(e.toString());
            }
        });
    }

12. In a try block in your implementation of the stop method, stop recording audio.
    RecordControl.commit().
    \_rcontrol.commit()

13. In the same try block you used in the previous step, write the audio data from the OutputStream to an array.
    ByteArrayOutputStream.toByteArray(). Store the value that
    ByteArrayOutputStream.toByteArray() returns inside the byte array.
    \_data = \_output.toByteArray();

14. In the same try block you used in the previous step, close the OutputStream and Player.
    ByteArrayOutputStream.close() and Player.close().
    \_output.close();
    \_player.close();

15. In a catch block, specify that a notification dialog displays with a message about an exception, if one occurs.
    } catch (Exception e) { 
        synchronized (UiApplication.getEventLock()) {
            Dialog.inform(e.toString());
        }
Code sample: recording audio from a media application

```java
final class VoiceNotesRecorderThread extends Thread
{

    private Player _player;
    private RecordControl _rcontrol;

    private ByteArrayOutputStream _output;
    private byte _data[];

    VoiceNotesRecorderThread() {}

    private int getSize()
    {
        return (_output != null ? _output.size() : 0);
    }

    private byte[] getVoiceNote()
    {
        return _data;
    }

    public void run()
    {
        try {
        ```
// Create a Player that records live audio.
_player = Manager.createPlayer("capture://audio");
_player.realize();

// Get the RecordControl, configure the record stream,
_rcontrol = (RecordControl)_player.getControl("RecordControl");

// Create a ByteArrayOutputStream to record the audio stream.
_output = new ByteArrayOutputStream();
_rcontrol.setRecordStream(_output);
_rcontrol.startRecord();
_player.start();

} catch (final Exception e) {
    UiApplication.getUiApplication().invokeAndWait(new Runnable() {
        public void run() {
            Dialog.inform(e.toString());
        }
    });
}

public void stop() {
    try {
        // Stop recording, record data from the OutputStream,
        // close the OutputStream and player.
        _rcontrol.commit();
        _data = _output.toByteArray();
        _output.close();
    }
}
_player.close();

} catch (Exception e) {
    synchronized (UiApplication.getEventLock()) {
        Dialog.inform(e.toString());
    }
}

1. The BlackBerry smartphone supports which of the following audio recording formats? Choose all that apply.
   A. 16 kHz mono-16-bit pulse code modulation (PCM)
   B. 8 kHz mono-16-bit pulse code modulation (PCM)
   C. Adaptive Multi-Rate (AMR)

2. When recording audio, you can use which of the following commands to configure the output stream where the application records data? Choose all that apply.
   A. _rcontrol.setRecordStream(_output);
   B. _rcontrol.setCaptureStream(_output);
   C. _rcontrol.setAudioInputStream(_output);
Answers

1. A or C

2. A
Supported audio and video formats

In the media application on the BlackBerry smartphone, you can open media files including videos, ring tones, pictures, and music files that are stored in the BlackBerry smartphone memory or on a media card.

BlackBerry Browser audio and video support

This section lists supported media formats by software versions. For a list of the formats supported on each smartphone, see “BlackBerry smartphone audio and video support” on page 221.

Support for media streaming

On BlackBerry smartphones with BlackBerry Device Software version 4.3.0 or later installed, the BlackBerry Browser supports RTSP streaming of audio and video files. The BlackBerry Browser supports media streaming over the HTTP or RTSP protocols.

When a BlackBerry smartphone user clicks a link to an audio or video file, the user can open the file, save it to the media card or device memory, or cancel the download. If the user opens the file, the BlackBerry Browser opens the media application and the file begins streaming. When the content is finished streaming, the user can close the media application to return to the browser. Streamed content is not saved. To replay media, users must download it again.

The network gateway can limit the size of the file that can be streamed. There are no size limitations if the content is streamed over a Wi-Fi® connection.
## Audio formats supported by the BlackBerry Browser

<table>
<thead>
<tr>
<th>Audio format</th>
<th>Supported codecs</th>
<th>RTSP Streaming</th>
<th>Supported by BlackBerry Device Software version</th>
</tr>
</thead>
<tbody>
<tr>
<td>3GP, 3G2</td>
<td>AAC-LC, AAC+, eACC+</td>
<td>Yes (4.5 or later)</td>
<td>4.5 or later</td>
</tr>
<tr>
<td></td>
<td>AMR-NB</td>
<td>Yes (4.5 or later)</td>
<td>4.5 or later</td>
</tr>
<tr>
<td></td>
<td>QCELP EVRC</td>
<td>No</td>
<td>4.7 or later</td>
</tr>
<tr>
<td>ASG=F</td>
<td>Windows Media Audio 9</td>
<td>Yes (4.7 or later)</td>
<td>4.3 or later</td>
</tr>
<tr>
<td></td>
<td>Windows Media Audio 10 Standard</td>
<td>Yes (4.7 or later)</td>
<td>4.3 or later</td>
</tr>
<tr>
<td></td>
<td>Windows Media Audio 10 Pro</td>
<td>Yes (4.7 or later)</td>
<td>4.3 or later</td>
</tr>
<tr>
<td>AVI</td>
<td>MP3</td>
<td>No</td>
<td>4.2 or later</td>
</tr>
<tr>
<td>MP3</td>
<td>MP3</td>
<td>No</td>
<td>4.2 or later</td>
</tr>
<tr>
<td>MP4, M4A</td>
<td>AAC-LC, AAC+, eACC+</td>
<td>Yes (4.5 or later)</td>
<td>4.2 or later</td>
</tr>
<tr>
<td></td>
<td>AMR-NB</td>
<td>Yes (4.5 or later)</td>
<td>4.2 or later</td>
</tr>
<tr>
<td></td>
<td>QCELP EVRC</td>
<td>Yes (4.7 or later)</td>
<td>4.7 or later</td>
</tr>
<tr>
<td>.MOV</td>
<td>AAC-LC, AAC+, eACC+</td>
<td>Yes (4.6 or later)</td>
<td>4.6 or later</td>
</tr>
<tr>
<td></td>
<td>AMR-NB</td>
<td>Yes (4.6 or later)</td>
<td>4.6 or later</td>
</tr>
<tr>
<td></td>
<td>QCELP EVRC</td>
<td>No</td>
<td>4.7 or later</td>
</tr>
<tr>
<td>.WMA</td>
<td>Windows Media Audio 9</td>
<td>Yes (4.7 or later)</td>
<td>4.2 or later</td>
</tr>
<tr>
<td></td>
<td>Windows Media Audio 10 Standard</td>
<td>Yes (4.7 or later)</td>
<td>4.2 or later</td>
</tr>
<tr>
<td></td>
<td>Windows Media Audio 10 Pro</td>
<td>Yes (4.7 or later)</td>
<td>4.2 or later</td>
</tr>
</tbody>
</table>
Video formats supported by the BlackBerry Browser

<table>
<thead>
<tr>
<th>Video container</th>
<th>Supported codecs</th>
<th>RTSP Streaming</th>
<th>Supported in</th>
</tr>
</thead>
<tbody>
<tr>
<td>3GP, 3GP2</td>
<td>H.264</td>
<td>Yes (4.6 or later)</td>
<td>4.6 or later</td>
</tr>
<tr>
<td></td>
<td>MPEG4</td>
<td>Yes (4.6 or later)</td>
<td>4.6 or later</td>
</tr>
<tr>
<td></td>
<td>H.263</td>
<td>Yes (4.6 or later)</td>
<td>4.6 or later</td>
</tr>
<tr>
<td>ASF</td>
<td>Windows Media Video 9</td>
<td>No</td>
<td>4.6 or later</td>
</tr>
<tr>
<td>AVI</td>
<td>MPEG4</td>
<td>Yes (4.6 or later)</td>
<td>4.6 or later</td>
</tr>
<tr>
<td>MP4, M4A</td>
<td>H.264</td>
<td>Yes (4.6 or later)</td>
<td>4.6 or later</td>
</tr>
<tr>
<td></td>
<td>MPEG4</td>
<td>Yes (4.6 or later)</td>
<td>4.6 or later</td>
</tr>
<tr>
<td></td>
<td>H.263</td>
<td>Yes (4.6 or later)</td>
<td>4.6 or later</td>
</tr>
<tr>
<td>MOV</td>
<td>H.264</td>
<td>Yes (4.6 or later)</td>
<td>4.6 or later</td>
</tr>
<tr>
<td></td>
<td>MPEG4</td>
<td>Yes (4.6 or later)</td>
<td>4.6 or later</td>
</tr>
<tr>
<td></td>
<td>H.263</td>
<td>Yes (4.6 or later)</td>
<td>4.6 or later</td>
</tr>
<tr>
<td>WMV</td>
<td>Windows Media Video 9</td>
<td>No</td>
<td>4.3 or later</td>
</tr>
</tbody>
</table>

BlackBerry smartphone audio and video support

This section lists supported media formats by software versions. For a list of the formats supported on each smartphone, see “BlackBerry Browser audio and video support” on page 219.

Specific codecs are recommended and supported for each file format for video and audio on each type of BlackBerry smartphone. Some formats support RTSP. Streaming requires BlackBerry Device Software version 4.3.0 or later.
## BlackBerry Storm 9500 smartphone and BlackBerry Storm 9530 smartphone

### Supported formats

<table>
<thead>
<tr>
<th>Format / Extension</th>
<th>Component</th>
<th>Codec</th>
<th>Notes</th>
<th>RTSP Streaming</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP4</td>
<td>Video</td>
<td>H.264</td>
<td>Baseline Profile, 480x360 pixels, up to 2 Mbps, 30 frames per second</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>Video</td>
<td>MPEG4</td>
<td>Simple Profile Level 3, 480x360 pixels, up to 2 Mbps, 30 frames per second</td>
<td>Supported</td>
</tr>
<tr>
<td>M4A</td>
<td>Video</td>
<td>H.263</td>
<td>Profile 0 and 3, Level 30</td>
<td>Supported</td>
</tr>
<tr>
<td>3GP</td>
<td>Audio</td>
<td>AAC-LC, AAC+, eAAC+</td>
<td>Supported</td>
<td></td>
</tr>
<tr>
<td>3GP2</td>
<td>Audio</td>
<td>AMR-NB</td>
<td>Supported</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Audio</td>
<td>QCELP EVRC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVI</td>
<td>Video</td>
<td>MPEG4</td>
<td>Simple Profile Level 3, 480x360 pixels, up to 2 Mbps, 30 frames per second</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>Audio</td>
<td>MP3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASF</td>
<td>Video</td>
<td>Windows Media Video 9</td>
<td>WMV3, Simple Profile, 480x360 pixels, 30 frames per second</td>
<td>Supported</td>
</tr>
<tr>
<td>WMV</td>
<td>Audio</td>
<td>Windows Media Audio 9</td>
<td>Supported</td>
<td></td>
</tr>
<tr>
<td>WMA</td>
<td>Audio</td>
<td>Windows Media 10 Standard/Professional</td>
<td>Supported</td>
<td></td>
</tr>
<tr>
<td>MP3</td>
<td>Audio</td>
<td>MP3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Recommended video formats for local playback

<table>
<thead>
<tr>
<th>Format / Extension</th>
<th>Component</th>
<th>Codec</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP4</td>
<td>Video</td>
<td>H.264</td>
<td>Baseline Profile, 480x360 pixels, up to 2 Mbps, 30 frames per second</td>
</tr>
<tr>
<td></td>
<td>Audio</td>
<td>AAC-LC</td>
<td></td>
</tr>
</tbody>
</table>

### BlackBerry Curve 8900 smartphone

#### Supported formats

<table>
<thead>
<tr>
<th>Format / Extension</th>
<th>Component</th>
<th>Codec</th>
<th>Notes</th>
<th>RTSP Streaming</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP4</td>
<td>Video</td>
<td>H.264</td>
<td>Baseline Profile, 480x360 pixels, up to 1500 kbps, 24 frames per second</td>
<td>Supported</td>
</tr>
<tr>
<td>M4A</td>
<td>Video</td>
<td>MPEG4</td>
<td>Simple and Advance Simple Profile, 480x360 pixels, up to 1500 kbps, 24 frames per second</td>
<td>Supported</td>
</tr>
<tr>
<td>MOV</td>
<td>Video</td>
<td>H.263</td>
<td>Profile 0 and 3, Level 45</td>
<td>Supported</td>
</tr>
<tr>
<td>MOV</td>
<td>Audio</td>
<td>AAC-LC, AAC+, eAAC+</td>
<td>Supported</td>
<td></td>
</tr>
<tr>
<td>MOV</td>
<td>Audio</td>
<td>AMR-NB</td>
<td>Supported</td>
<td></td>
</tr>
<tr>
<td>3GP</td>
<td>Video</td>
<td>MPEG4</td>
<td>Simple and Advance Simple Profile, 480x360 pixels, up to 1500 kbps, 24 frames per second</td>
<td>Supported</td>
</tr>
<tr>
<td>3GP</td>
<td>Audio</td>
<td>MP3</td>
<td></td>
<td>Supported</td>
</tr>
<tr>
<td>Format / Extension</td>
<td>Component</td>
<td>Codec</td>
<td>Notes</td>
<td>RTSP Streaming</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>--------------------------------</td>
<td>--------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>ASF</td>
<td>Video</td>
<td>Windows Media Video 9</td>
<td>WMV3, Simple and Main Profile, 480x360 pixels, 24 frames per second</td>
<td></td>
</tr>
<tr>
<td>WMV</td>
<td>Audio</td>
<td>Windows Media 9 Standard/Professional</td>
<td></td>
<td>Supported</td>
</tr>
<tr>
<td>WMA</td>
<td>Audio</td>
<td>Windows Media 10 Standard/Professional</td>
<td></td>
<td>Supported</td>
</tr>
<tr>
<td>MP3</td>
<td>Audio</td>
<td>MP3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Recommended video formats for local playback**

<table>
<thead>
<tr>
<th>Format / Extension</th>
<th>Component</th>
<th>Codec</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP4</td>
<td>Video</td>
<td>MPEG4</td>
<td>Advance Simple Profile, 480x360 pixels, up to 1500 kbps, 24 frames per second</td>
</tr>
<tr>
<td></td>
<td>Audio</td>
<td>AAC-LC</td>
<td></td>
</tr>
</tbody>
</table>

**BlackBerry Bold 9000 smartphone**

**Supported formats**

<table>
<thead>
<tr>
<th>Format / Extension</th>
<th>Component</th>
<th>Codec</th>
<th>Notes</th>
<th>RTSP Streaming</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP4</td>
<td>Video</td>
<td>H.264</td>
<td>Baseline Profile, 480x320 pixels, up to 1500 kbps, 24 frames per second</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>Video</td>
<td>MPEG4</td>
<td>Simple and Advance Simple Profile, 480x320 pixels, up to 1500 kbps, 24 frames per second</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>Video</td>
<td>H.263</td>
<td>Profile 0 and 3, Level 45</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>Audio</td>
<td>AAC-LC, AAC+, eAAC+</td>
<td></td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>Audio</td>
<td>AMR-NB</td>
<td></td>
<td>Supported</td>
</tr>
<tr>
<td>MOV</td>
<td>Video</td>
<td>MPEG4</td>
<td>Simple and Advance Simple Profile, 480x320 pixels, up to 1500 kbps, 24 frames per second</td>
<td>Supported</td>
</tr>
<tr>
<td>3GP</td>
<td>Audio</td>
<td>MP3</td>
<td></td>
<td>Supported</td>
</tr>
<tr>
<td>Format / Extension</td>
<td>Component</td>
<td>Codec</td>
<td>Notes</td>
<td>RTSP Streaming</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------</td>
<td>------------------------</td>
<td>--------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>ASF</td>
<td>Video</td>
<td>Windows Media Video 9</td>
<td>WMV3, Simple and Main Profile, 480x320 pixels, 24 frames per second</td>
<td></td>
</tr>
<tr>
<td>WMV</td>
<td>Audio</td>
<td>Windows Media 9</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Standard/Professional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WMA</td>
<td>Audio</td>
<td>Windows Media 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Standard/Professional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MP3</td>
<td>Audio</td>
<td>MP3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Recommended video formats for local playback

<table>
<thead>
<tr>
<th>Format / Extension</th>
<th>Component</th>
<th>Codec</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP4</td>
<td>Video</td>
<td>MPEG4</td>
<td>Advance Simple Profile, 480x320 pixels, up to 1500 kbps, 24 frames per second</td>
</tr>
<tr>
<td></td>
<td>Audio</td>
<td>AAC-LC</td>
<td></td>
</tr>
</tbody>
</table>

BlackBerry 8820 smartphone (GSM/GPRS/EDGE networks)

Supported formats

<table>
<thead>
<tr>
<th>Format / Extension</th>
<th>Component</th>
<th>Codec</th>
<th>Notes</th>
<th>RTSP Streaming</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP4</td>
<td>Video</td>
<td>MPEG4</td>
<td>Simple Profile, 320x240 pixels, up to 768 kbps, 24 frames per second</td>
<td>Supported</td>
</tr>
<tr>
<td>M4A</td>
<td>Video</td>
<td>H.263</td>
<td>Profile 0 and 3, Level 45</td>
<td>Supported</td>
</tr>
<tr>
<td>MOV</td>
<td>Audio</td>
<td>AAC-LC, AAC+, eAAC+</td>
<td></td>
<td>Supported</td>
</tr>
<tr>
<td>3GP</td>
<td>Audio</td>
<td>AMR-NB</td>
<td></td>
<td>Supported</td>
</tr>
<tr>
<td>AVI</td>
<td>Video</td>
<td>MPEG4</td>
<td>Simple and Advance Simple Profile, 320x240 pixels, up to 768 kbps, 24 frames per second</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>Audio</td>
<td>MP3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASF</td>
<td>Video</td>
<td>Windows Media Video 9</td>
<td>WMV3, Simple Profile, 320x240 pixels, 24 frames per second</td>
<td></td>
</tr>
<tr>
<td>WMV</td>
<td>Audio</td>
<td>Windows Media 9 Standard/Professional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WMA</td>
<td>Audio</td>
<td>Windows Media 10 Standard/Professional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MP3</td>
<td>Audio</td>
<td>MP3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Recommended video format for local playback (BlackBerry Device Software version 4.5.0)

<table>
<thead>
<tr>
<th>Format / Extension</th>
<th>Component</th>
<th>Codec</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP4</td>
<td>Video</td>
<td>MPEG4</td>
<td>Simple Profile, 320x240 pixels, up to 768 kbps, 24 frames per second</td>
</tr>
<tr>
<td></td>
<td>Audio</td>
<td>AAC-LC</td>
<td></td>
</tr>
</tbody>
</table>

Recommended video format for local playback (BlackBerry Device Software version 4.2.0 and 4.3.0)

<table>
<thead>
<tr>
<th>Format / Extension</th>
<th>Component</th>
<th>Codec</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP4</td>
<td>Video</td>
<td>MPEG4</td>
<td>Simple Profile, 320x240 pixels, up to 768 kbps, 24 frames per second</td>
</tr>
<tr>
<td></td>
<td>Audio</td>
<td>MP3</td>
<td></td>
</tr>
</tbody>
</table>
Quiz

1. On BlackBerry smartphones with BlackBerry Device Software version 4.2.0 or later installed, the BlackBerry Browser supports which of the following web feed formats? Choose all that apply.
   A. RSS 0.9, 1.0, and 2.0
   B. Windows Video
   C. Atom™ RSS 0.9, 1.0, and 2.0
   D. ANIM 6.0, 7.0

2. On BlackBerry smartphones with BlackBerry Device Software version 4.3.0 or later installed, the BlackBerry Browser supports which of the following audio types?
   A. MIDI
   B. MP3
   C. MP4
   D. WAV
   E. all of the above
   F. none of the above

3. On BlackBerry smartphones with BlackBerry Device Software version 4.3.0 or later installed, the BlackBerry Browser supports which of the following video types?
   A. AVI
   B. WMV
   C. FLC
   D. h.263
   E. all of the above
   F. none of the above
Answers

1. A and C
2. E
3. B and D
You can create a BlackBerry device application that plays media in the BlackBerry Browser or in the media application on a BlackBerry smartphone. You can also create a BlackBerry smartphone media application that can play audio, video, and binary SVG content, record audio and video, or send audio to a Bluetooth enabled headset.
1. List the web feed formats that are supported on BlackBerry smartphones with BlackBerry Device Software version 4.2.0 or later installed.

2. List the streaming media formats that are supported by BlackBerry smartphones with BlackBerry Device Software version 4.3.0 or later installed.

3. List the classes used to play media in the BlackBerry Browser.

4. You can use the Multimedia API to create a custom class that extends what protocol to buffer and play streaming media?

5. What JSR can you use to create a media application?