Advanced Java Application Development for the BlackBerry Smartphone Lab manual
All other trademarks are the property of their respective owners.

The BlackBerry smartphone and other devices 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 the U.S. and in various countries around the world. Visit www.rim.com/patents for a list of RIM (as hereinafter defined) patents.

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

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

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

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW IN YOUR JURISDICTION, IN NO EVENT SHALL RIM BE LIABLE FOR ANY TYPE OF DAMAGES RELATED TO THIS DOCUMENTATION OR ITS USE, OR PERFORMANCE OR NON-PERFORMANCE OF ANY SOFTWARE, HARDWARE, SERVICE, OR ANY THIRD PARTY PRODUCTS AND SERVICES REFERENCED HEREIN INCLUDING WITHOUT LIMITATION ANY OF THE FOLLOWING DAMAGES: DIRECT, CONSEQUENTIAL, EXEMPLARY, INCIDENTAL, INDIRECT, SPECIAL, PUNITIVE, OR AGGRAVATED DAMAGES, DAMAGES FOR LOSS OF PROFITS OR REVENUES, FAILURE TO
REALIZE ANY EXPECTED SAVINGS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, LOSS OF BUSINESS OPPORTUNITY, OR CORRUPTION OR LOSS OF DATA, FAILURES TO TRANSMIT OR RECEIVE ANY DATA, PROBLEMS ASSOCIATED WITH ANY APPLICATIONS USED IN CONJUNCTION WITH RIM PRODUCTS OR SERVICES, DOWNTIME COSTS, LOSS OF THE USE OF RIM PRODUCTS OR SERVICES OR ANY PORTION THEREOF OR OF ANY AIRTIME SERVICES, COST OF SUBSTITUTE GOODS, COSTS OF COVER, FACILITIES OR SERVICES, COST OF CAPITAL, OR OTHER SIMILAR PECCUNIARY LOSSES, WHETHER OR NOT SUCH DAMAGES WERE FORESEEN OR UNFORESEEN, AND EVEN IF RIM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

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

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

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

Prior to subscribing for, installing or using any Third Party Products and Services it is your responsibility to ensure that your airtime service provider has agreed to support all of their features. Some airtime service providers 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 software, BlackBerry® Desktop Software, and/or BlackBerry® Device Software and may require additional development or Third Party Products and Services for access to corporate applications.

This product includes 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

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.
5.1 Access media and create player

In this exercise, you will learn how to initialize playable media on the BlackBerry smartphone and create a player. This lab shows you how to use the EmbeddedMediaScreen class to access media, create a Player object and implement the video and volume controls.

10. Open the EmbeddedMediaScreen.java source file.

11. Create an InputStream object from the bundled media file in the initializeMedia() method of the EmbeddedMediaScreen class.

   InputStream is = getClass().getResourceAsStream("/BlackBerry.mp4");

12. Create a media player by calling a static method of the javax.microedition.media.Manager class to return a Player object. Do this in the initializeMedia() method of the EmbeddedMediaScreen class.

   _player = Manager.createPlayer(is, "video/mp4");

13. Transition the Player object from the Unrealized to Realized state in the initializeMedia() method of the EmbeddedMediaScreen class. A Player object is in the Realized state when it has obtained the information it requires to acquire the media resource.

   _player.realize();

14. Create a VideoControl object in the initializeMedia() method of the EmbeddedMediaScreen class. You can obtain a VideoControl object by passing the value VideoControl to the getControl() method.

   VideoControl videoControl = (VideoControl)_player.getControl("VideoControl");

15. In the initializeMedia() method of the EmbeddedMediaScreen class, call the initDisplayMode() method of the VideoControl interface to display video in a standard BlackBerry UI field. To do this you need to do the following,

   a. Set the video mode to USE_GUI_PRIMITIVE.

   b. Pass net.rim.device.api.ui.Field as the second argument to the method.

      _videoField = (Field) videoControl.initDisplayMode
                  (VideoControl.USE_GUI_PRIMITIVE,
                   "net.rim.device.api.ui.Field");
16. Make the video control visible on the screen in the initializeMedia() method of the EmbeddedMediaScreen class.
   videoControl.setVisible(true);

17. Call the same method you used to return the VideoControl, only this time we specify that the method should return a VolumeControl object.
   _volumeControl = (VolumeControl)_player.getControl("VolumeControl");
5.2 Starting and stopping media

In this exercise, you will learn how to implement a button you can use to start and stop media in a player. You will implement this code in the fieldChanged() method of the EmbeddedMediaScreen class in this example.

1. Open the EmbeddedMediaScreen.java source file.

2. The Player interface provides methods to start and stop the playing of the media. A MediaException is thrown if the video cannot be started or stopped.

Note: When calling start(), the media resumes playback from the point where stop() was called. To reset the video, call the Player setMediaTime() method to reset the time to 0.

```java
if (_player.getState() != Player.STARTED) {
    _player.start();
    _timerUpdateThread = new TimerUpdateThread();
    _timerUpdateThread.start();
} else {
    _player.stop();
    _timerUpdateThread.stop();
}
```
Chapter 5: Introduction to multimedia application development for the BlackBerry smartphone

Notes
5.3 Adjusting volume

In this exercise, you will learn how to adjust the volume during media playback. You can accomplish this by overriding the keyControl() method of the EmbeddedMediaScreen class to intercept key press events.

1. Open the EmbeddedMediaScreen.java source file.

2. Invoke the setLevel() method, passing a volume level reduced by 10 as the argument. If the new level is less than zero, the method automatically sets the new level to 0.
   
   _volumeControl.setLevel(_volumeControl.getLevel() - 10);

3. Invoke the setLevel() method, passing a volume level increased by 10 as the argument. If the new level is greater than 100, the method sets the new level to 100.
   
   _volumeControl.setLevel(_volumeControl.getLevel() + 10);
5.4 Implementing a PlayerListener

In this exercise, you will learn how to implement the EmbeddedMediaScreen class as a PlayerListener, and add code to respond to Player events. The PlayerListener interface requires you to implement the playerUpdated() method. This method is called whenever _player generates an event (STARTED, STOPPED, VOLUME_CHANGED, and so on.).

1. Open the EmbeddedMediaScreen.java source file.

2. Register the class as a PlayerListener in the initializeMedia() method.
   _player.addPlayerListener(this);

3. When the volume of the player is changed, update the display to reflect this change. This action is handled by placing code in the playerUpdate() method that changes the label text.
   _volumeDisplay.setText("Volume : "+ _volumeControl.getLevel());

4. When the player is started, the label on the play button changes to Pause so that it can be used to pause the media file. This action is handled by again placing code in the playerUpdate() method.
   _controlButton.setLabel("Pause");

5. When the player is paused, the label on the play button changes to Start so that the button can be used to play the media file. This action is handled by again placing code in the playerUpdate() method.
   _controlButton.setLabel("Start");

6. You cannot determine the length of a media file ahead of time. The DURATION_UPDATED event is generated when the duration becomes available. Update the _duration LabelField to display the duration of the media content in seconds by placing code in the playerUpdate() method.
   _duration.setText(_player.getDuration()/1000000 + " s");

7. When the end of the media file is encountered, the label on the play button changes so it can be used to start the file again. This action is accomplished by placing code in the playerUpdate() method.
   _controlButton.setLabel("Start");
5.5 Update the time elapsed

In this exercise you will learn how to update the time elapsed display for the media player.

1. Open the `EmbeddedMediaScreen.java` source file.

2. Update the label displaying the time elapsed using the `TimerUpdateThread` class `run()` method by doing the following:
   a. Obtain the current media time of the media file using `getMediaTime()` in the `Player` class.
   b. Update the `_currentTime` LabelField to display the media time in seconds. Note that `getMediaTime()` returns the value in microseconds.

   ```java
   _currentTime.setText( _player.getMediaTime() / 1000000 + ";" );
   ```
5.6 Build and run the project

In this exercise you will learn how to build and run this project in the BlackBerry Smartphone Simulator.

1. In the Package Explorer view, right-click on the Lab_08_MobileMedia project and click Activate for BlackBerry from the menu. The project folder icon changes indicating that the project is activated.

2. In the Package Explorer view, right-click on the Lab_08_MobileMedia project and click Run As from the menu.

3. Select BlackBerry Simulator.

4. When the BlackBerry Smartphone Simulator Home screen appears, press the Menu key to display all applications and folders.

5. Use the arrow keys on your keyboard to navigate to the Downloads folder.

6. Enter the folder by pressing the ENTER key on your keyboard.

7. Use the arrow keys on your keyboard to navigate to the generic application Mobile Media Lab icon.

8. Invoke the application by pressing the ENTER key on your keyboard.