

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV](#) [NEXT](#)[FRAMES](#) [NO FRAMES](#)

Multimedia

Overview

This document describes how to use the multimedia capabilities of the BlackBerry® device. It includes the following sections:

- [The media player](#)
- [Supported audio formats](#)
- [Play audio](#)
- [Record audio](#)
- [Play video](#)
- [Record video](#)

The media player [\[back to top\]](#)

To record and play media in your BlackBerry device application, you must use the [Player](#) class, which is located in the [javax.microedition.media](#) package, and its related resources.

To create an instance of a Player class, you must invoke [createPlayer\(\)](#) in the [javax.microedition.media.Manager](#) class.

When the Player is first created using one of the [createPlayer\(\)](#) methods, it is in an UNREALIZED state. Before you can have access to the associated metadata or controls for the Player, you must invoke the Player object's [realize\(\)](#) method. This method transitions the Player to a REALIZED state.

In the REALIZED state, the Player class provides access to associated resources that control playback and recording. You can use the Player object's [getControl\(\)](#) and [getControls\(\)](#) methods (inherited from [Controllable](#)), passing in a String specifying the [Control](#) object that you require. You can find the available controls for your media player in the [javax.microedition.media.control](#) package.

Before your application closes, you must invoke the Player object's [close\(\)](#) method to deallocate any resources the Player has created. You can also do this at any time to free up memory or resources. After you invoke [close\(\)](#), you can no longer use the Player.

For more information about the states of a Player object and how to transition between them, see the API reference for the Player class.

Supported audio formats [\[back to top\]](#)

Before you begin playing and recording audio, here is a list of audio formats supported by most BlackBerry devices running OS 4.2 or later:

- WAV (PCM)
- MP3
- AMR (AMR/AMR-NB)

- AAC (AAC+, AAC-LC, eAAC+) (except for the BlackBerry® 8700 Series)
- WMA (v9 and v10 Pro) (except for some models from the BlackBerry 8700 Series, BlackBerry® 8800 Series, BlackBerry® 8100 Series, and BlackBerry® 8300 Series)
- MIDI (GM, GM2, SP-MIDI)
- SP-MIDI

BlackBerry devices that are running BlackBerry Device Software 5.0 or later also support the following audio formats:

- FLAC
- Vorbis

Play audio [\[back to top\]](#)

1. Create a [Player](#) object by invoking the [createPlayer\(String\)](#) method in the [javax.microedition.media](#) class, passing in the location of the audio file to play.
2. Invoke the Player object's [realize\(\)](#) method to access the [Control](#) resources. Note that invoking [Player.start\(\)](#) implicitly performs all necessary state transitions. In this example, we explicitly invoke both [realize\(\)](#) and [prefetch\(\)](#) to demonstrate how to explicitly initialize the Player object before starting playback.
3. To control an aspect of the playback, retrieve the appropriate [Control](#) object. The following code sample demonstrates how to retrieve the [VolumeControl](#) control to set the volume level.
4. To start the playback, invoke the Player object's [start\(\)](#) method.
5. To pause the playback, invoke the Player object's [stop\(\)](#) method.

Before your application closes, you must invoke the Player object's [close\(\)](#) method to deallocate any resources the Player has created.

 Click for code sample: Playing an audio file

Record audio [\[back to top\]](#)


To start recording audio on a BlackBerry device using the device's microphone, follow these steps:

1. Create a new [Player](#) object specifying the appropriate protocol for file encoding.
2. Invoke the Player object's [realize\(\)](#) method to access the Control resources.
3. To access the recording functionality, retrieve the [RecordControl](#) object associated with the Player object.
4. To set the location of the recorded file, invoke the RecordControl object's [setRecordLocation\(\)](#) or [setRecordStream\(\)](#) method.
5. Start recording by invoking the RecordControl objects' [startRecord\(\)](#) method and the Player object's [start\(\)](#) method.

 Click for code sample: Starting recording audio to a file

To stop recording and save your file, follow these steps:

1. To release the Player object's resources, invoke the Player object's [close\(\)](#) method.
2. To stop the recording, invoke the RecordControl object's [stopRecord\(\)](#) method.
3. To save the recording to the specified file, invoke the RecordControl object's [commit\(\)](#) method.

 Click for code sample: Stopping an audio recording and saving it to a file

Play video [\[back to top\]](#)

1. Create a [Player](#) object by invoking the [createPlayer\(String\)](#) method in the [javax.microedition.media](#) class, passing in the location of the video file to play.
2. Invoke the Player object's [realize\(\)](#) method to access the [Control](#) resources.
3. Invoke the Player object's [getControl\(\)](#) method, passing in the String "VideoControl", to retrieve the [VideoControl](#) object that is associated with the Player.
4. Invoke the VideoControl object's [initDisplayMode\(\)](#) method. In the following code sample we will use the [USE_GUI_PRIMITIVE_MODE](#), and so the second parameter of [initDisplayMode\(\)](#) is a String that specifies the object that displays the video ("net.rim.device.api.ui.Field"). The [initDisplayMethod\(\)](#) method returns a Field object.
5. Invoke your Manager object's [add\(\)](#) method to add the returned Field object to your Screen or Manager. This is the same as adding any other component to your UI.
6. To control an aspect of the video playback, retrieve the appropriate Control object. The following code sample demonstrates how to set the volume level of the playback.
7. To start the playback, invoke the Player object's [start\(\)](#) method.
8. To pause the playback, invoke the Player object's [stop\(\)](#) method.

When your application closes, invoke the Player object's [close\(\)](#) method to deallocate any resources that the Player has created.

[Click for code sample: Playing a video file](#)

Record video [\[back to top\]](#)

To start recording video to a file in your BlackBerry application, follow these steps:

1. Create a new [Player](#) object specifying the appropriate protocol for file encoding.
2. Invoke the Player object's [realize\(\)](#) method to access the [Control](#) resources.
3. Invoke the Player object's [getControl\(\)](#) method, passing in the String "VideoControl", to retrieve the [VideoControl](#) object that is associated with this Player object.
4. Invoke the VideoControl object's [initDisplayMode\(\)](#) method. In the following code sample, we will use the [USE_GUI_PRIMITIVE_MODE](#), and so the second parameter of the [initDisplayMode\(\)](#) method will be a String specifying the Object displaying the video ("net.rim.device.api.ui.Field"). The [initDisplayMethod\(\)](#) returns a Field object.
5. Invoke your Manager object's [add\(\)](#) method to add the returned Field object to your Screen or Manager. This is the same as adding any component to your UI.
6. To access the recording functionality, retrieve the [RecordControl](#) object that is associated with the Player object.
7. To set the location of the recorded file, invoke the RecordControl object's [setRecordLocation\(\)](#) method.
8. Start recording by invoking the RecordControl object's [start\(\)](#) method and the Player object's [start\(\)](#) method.

[Click for code sample: Starting a video recording](#)

To stop recording and save your file, follow these steps:

1. To release the Player object's resources, invoke the Player object's [close\(\)](#) method.
2. To stop the recording, invoke the RecordControl object's [stopRecord\(\)](#) method.
3. To save the recording to the specified file, invoke the RecordControl object's [commit\(\)](#) method.

Click for code sample: Stopping a video recording and saving it

Copyright 1999-2009 Research In Motion Limited. 295 Phillip Street, Waterloo, Ontario, Canada, N2L 3W8. All Rights Reserved.

Copyright 1993-2003 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California, 94303, U.S.A.

Copyright 2002-2003 Nokia Corporation All Rights Reserved.

Java is a trademark or registered trademark of Sun Microsystems,