BlackBerry Java Development Environment
Version 3.6

Creating MIDP Applications
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About this guide</td>
<td>3</td>
</tr>
<tr>
<td>Overview</td>
<td>4</td>
</tr>
<tr>
<td>Audience</td>
<td>4</td>
</tr>
<tr>
<td>Conventions</td>
<td>4</td>
</tr>
<tr>
<td>Related resources</td>
<td>5</td>
</tr>
<tr>
<td>1 Working with MIDlets</td>
<td>7</td>
</tr>
<tr>
<td>Java support on BlackBerry handhelds</td>
<td>8</td>
</tr>
<tr>
<td>Creating MIDlets</td>
<td>10</td>
</tr>
<tr>
<td>Creating MIDlet suites</td>
<td>10</td>
</tr>
<tr>
<td>Deploying MIDlets</td>
<td>11</td>
</tr>
<tr>
<td>2 Designing MIDlets</td>
<td>13</td>
</tr>
<tr>
<td>Designing MIDlets</td>
<td>14</td>
</tr>
<tr>
<td>Using key mappings for games</td>
<td>15</td>
</tr>
<tr>
<td>3 Comparing MIDP and BlackBerry UI APIs</td>
<td>17</td>
</tr>
<tr>
<td>BlackBerry and MIDP APIs</td>
<td>18</td>
</tr>
<tr>
<td>Comparison of UI components</td>
<td>20</td>
</tr>
</tbody>
</table>
About this guide

This section provides information on the following topics:

- Overview
- Audience
- Conventions
- Related resources
About this guide

Overview
This guide explains how to create MIDP 1.0 applications, called MIDlets, that work well on BlackBerry handhelds. You can find information on the following topics:

- how to convert MIDlet applications into the BlackBerry format
- how to design a MIDlet user interface that is optimized for BlackBerry handhelds
- how to understand the relationship between MIDP and BlackBerry UI APIs

Product compatibility
Unless otherwise noted, the information in this book applies to the following BlackBerry products:

<table>
<thead>
<tr>
<th>Product</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlackBerry Java Development Environment</td>
<td>3.6 or later</td>
</tr>
<tr>
<td>BlackBerry handheld software</td>
<td>3.6 or later</td>
</tr>
<tr>
<td>BlackBerry Enterprise Server for Microsoft® Exchange</td>
<td>3.6 or later</td>
</tr>
<tr>
<td>BlackBerry Enterprise Server for Lotus® Domino™</td>
<td>2.2 or later</td>
</tr>
</tbody>
</table>

Audience
This guide is intended for Java software developers who want to develop and deploy MIDP 1.0 compliant applications for BlackBerry handhelds.

This book is not intended to teach basic Java™ or wireless programming. Readers should already know how to program using standard Java programming, including networking, and should also have some basic knowledge of wireless programming and the BlackBerry solution.

Conventions

⚠️ **Warning:** Warnings advise you that failure to take or avoid a specific action could result in data loss or physical damage.

⚠️ **Note:** Notes contain important information on the associated topic.

⚠️ **Tip:** Tips provide optional or time-saving information on the associated topic.
Related resources
These other resources can help you to develop wireless applications for BlackBerry.

Background information
For information on Java™ 2 Platform, Micro Edition (J2ME™), including the Connected Limited Device Configuration (CLDC) and Mobile Information Device Profile (MIDP) specifications, refer to the Sun Microsystems™ web site:
http://java.sun.com/j2me/
http://java.sun.com/products/midp
http://java.sun.com/products/cldc

Related documentation
• BlackBerry Java Developer Guide Volume 1 – Fundamentals
  Volume 1 provides information on the BlackBerry solution, application design considerations, and development tools, as well as information on BlackBerry APIs for user interfaces, localization, and networking.
• Online Help Title
  The Online Help Title provides detailed procedures for using the integrated development environment (IDE). To view the IDE Online Help in the IDE, on the Help menu, click IDE Help. Use the table of contents, index, or full-text search to find specific information.
• API Reference
  The online API reference documentation provides detailed descriptions of the APIs that are provided with the BlackBerry Java Development Environment. To view API documentation in the IDE, on the Help menu, click API Reference. The API reference documentation is also available at http://www.blackberry.com/developers.

Developer support
Visit the BlackBerry Developer Zone at http://www.blackberry.com/developers for additional documentation, technical updates, tutorials, and a developer discussion forum.
About this guide
Working with MIDlets

This section provides information on the following topics:

• Java support on BlackBerry handhelds
• Creating MIDlets
• Creating MIDlet suites
Java support on BlackBerry handhelds

BlackBerry™ handhelds support Java™ 2 Platform Micro Edition (J2ME™) as well as BlackBerry API extensions.

BlackBerry JDE

The BlackBerry Java Development Environment (JDE) includes a complete set of APIs and tools that you can use to develop custom Java applications for the BlackBerry Wireless Handheld.

The Product Name Short includes the following items:

- IDE
- BlackBerry simulation tools
- Mobile Data Service simulation tools
- CLDC, MIDP, and BlackBerry APIs
- sample applications

The IDE includes a full suite of editing and debugging tools that are optimized for developing BlackBerry applications. In addition to several advanced debugging features, the IDE includes simulation tools that provide a complete Windows-based environment in which you can simulate user interfaces, user interaction, and network conditions.

The IDE compiles and packages source code into .cod files, which can be loaded onto the handheld. You can use the IDE to convert MIDP applications (called MIDlets) into .cod file format.

APIs

The Product Name Short supports both the J2ME APIs in the Connected Limited Device Configuration (CLDC) and the Mobile Information Device Profile 1.0 (MIDP). The BlackBerry JDE also supports BlackBerry API extensions.

Applications that use standard MIDP 1.0 APIs, sometimes called MIDlets, can run on BlackBerry handhelds and other J2ME-enabled wireless devices. Applications that use BlackBerry APIs can take advantage of additional BlackBerry features, such as more sophisticated user interfaces, multithreading, language localization support, and more robust data storage.
Network transport

MIDlets can communicate with networks using standard HTTP connections, regardless of the underlying wireless network. On the BlackBerry handheld, when a MIDlet opens a standard HTTP connection (http://) a BlackBerry Enterprise Server, with the Mobile Data Service feature enabled, is required.

To open a network connection using a service provider's WAP gateway, an application must add additional parameters to the URL. Refer to the BlackBerry Java Developer Guide for more information.

In addition to the standard HTTP support in MIDP, the BlackBerry handheld supports secure HTTP (HTTPS) and socket connections.

Persistent storage

The BlackBerry handheld provides two ways to store data persistently:

- MIDP record management store (RMS)
- BlackBerry persistent store

Use the MIDP record store if you want your application to be portable across multiple devices that are compatible with the Java™ 2 Platform, Micro Edition (J2ME™). If you are writing an application specifically for the BlackBerry handheld, you typically use the BlackBerry persistence model because it provides a more flexible and efficient way to store data.

There are two key differences between the MIDP RecordStore and the BlackBerry PersistentStore:

- **Data storage**: MIDP records store data as byte arrays. The BlackBerry PersistentStore can store complete objects.
- **Data sharing**: In MIDP, each RecordStore belongs to a single MIDlet suite and a MIDlet can only access record stores that are created by a MIDlet in the same suite. In the BlackBerry PersistentStore, data can be shared between applications, at the discretion of the application that creates the data.

Refer to the BlackBerry Java Developer Guide for more information.
Creating MIDlets

MIDP applications, or MIDlets, are applications that use the CLDC and MIDP APIs only.

Create a MIDlet project
1. In the IDE, right-click the project.
2. Click Properties. The project properties window appears.
3. Click the Application tab.
4. From the Project type drop-down list, select MIDlet.
5. In the Name of main MIDlet class field, type the fully qualified name of the class in your MIDlet application that contains the startApp() method.

When you compile the project, the IDE creates a .cod file and .jar and .jad files. Load the .cod file on the BlackBerry handheld. On other MIDP-compatible devices, load the .jar and .jad files.

Creating MIDlet suites

A MIDlet suite consists of one or more MIDlet applications that are contained in a single .jar file. The MIDlets can share resources and classes.

All source files for a MIDlet must be contained in a single .cod file. The source files for a MIDlet suite cannot span multiple projects.

To create a MIDlet suite
1. Create a main project for all source files.
2. In the main project, set the project type to MIDlet.
3. Create a separate project for each MIDlet in the MIDlet suite.
4. In each separate project, set the project type to Alternate MIDlet Entry Point. Refer to "Set alternate entry points" below for more information.
Set alternate entry points

The IDE enables you to define multiple entry points for an application. For example, each MIDlet in a MIDlet suite is an entry point into the same application.

1. In the IDE, create a new project.
2. Add an application icon to the project. The project should not contain any other files.
3. Right-click the project file.
4. Click Properties. The project properties window appears.
5. Click the Application tab.
6. From the Project type drop-down list, select Alternate MIDlet Entry Point.
7. From the Alternate entry point for drop-down list, select the main project for the MIDlet suite.
8. In the Name of main MIDlet class field, type the fully qualified name of the class that extends the MIDlet class (for example, com.company.test.MidletTest).

When this MIDlet is invoked on the handheld, a new instance of this MIDlet class is created.
9. Click OK.

When you build and run the applications, this alternate entry point appears on the handheld Home screen as a separate application, but it actually starts the application for which it is an entry point.

Deploying MIDlets

With BlackBerry handheld software version 3.6 or later, users can download applications wirelessly using the BlackBerry Browser. MIDlet .jar files cannot be loaded onto a BlackBerry handheld. The .jar files must first be converted into the .cod file format that the BlackBerry handheld accepts.

The Mobile Data Service feature of the BlackBerry Enterprise Server converts .jar files to .cod files, so that users can download MIDlet .jar and .jad files. When a user requests a MIDlet to download, the Mobile Data Service converts the .jar file to a .cod file before sending it to the handheld.

The following versions of the BlackBerry Enterprise Server can convert .jar files to .cod files:

- BlackBerry Enterprise Server version 3.6 or later for Microsoft Exchange
- BlackBerry Enterprise Server version 2.2 or later for Lotus Domino

Note: Users can only download .jar files if they access the network using a BlackBerry Enterprise Server with the Mobile Data Service feature enabled. The Mobile Data Service converts .jar files to the .cod file format that the handheld requires. If users access the network using a WAP gateway, they can only download .cod files. Developers must convert MIDlet .jar files to .cod file format for users to download.
1: Working with MIDlets

Convert MIDlet .jar files to .cod files using the IDE

1. In the IDE, open a workspace or create a new workspace.
2. Create a new project for the MIDlet application.
3. Add the MIDlet .jad and .jar files to the project.

⚠️ Note: All code for a MIDlet application must be included in a single project. You can place other files, such as images, in a separate library project.

1. Right-click the project.
2. Click Activate Project. The project is now included in builds.
3. Right-click the project.
4. Click Properties. The project properties window appears.
5. On the General tab, type appropriate values for Title, Version, and Vendor.
   Values that are specified in the .jad file and .jar manifest override values that are specified in the IDE project properties.
6. Click the Application tab.
7. From the Project type drop-down list, select MIDlet.
8. In the Name of main MIDlet class field, type the fully qualified name of the class in your MIDlet application that contains the startApp() method.
9. Click OK.
10. On the Build menu, click Build. The IDE generates the necessary .cod files.
Designing MIDlets

This section provides information on the following topics:

- Designing MIDlets
- Using key mappings for games
2: Designing MIDlets

Designing MIDlets
This section provides guidelines for optimizing MIDlets for BlackBerry handhelds.

Painting
The Canvas.paint method should clear the entire screen. Some applications clear an area based on the anticipated screen size. The BlackBerry handheld has a larger screen than many other devices, such as cell phones.

The Canvas.paint method accepts a single parameter: a Graphics object with a clip region that defines the area of the screen that is considered to be invalid. Implementations of the paint method must redraw every pixel within this region. An application can display incorrectly if it paints only a subset of pixels within this region, and screen contents are invalidated by a repaint() request from an external source, such as the BlackBerry application framework.

Commands
Use the appropriate command types for your commands. The BlackBerry application framework relies on command types to prevent duplicate menu items.

In particular, if a command exits the application, use Command.EXIT as the type. On the BlackBerry platform, a Close item is added automatically to the menu, unless a command is specified as Command.EXIT, in which case that command is used instead. If the MIDlet does not use the appropriate types on its commands, the menu can contain both an Exit and a Close item.

Escape key
The handheld Escape button is mapped to Command.CANCEL, Command.BACK, or Command.EXIT commands, in that order. MIDlet applications should implement at least one of these commands to provide a response when the user presses the Escape button.

Phone soft keys
The handheld does not have the two phone soft keys that other mobile devices, such as cell phones, provide. Instead, commands that would be associated with phone soft keys are displayed as items in the application menu.

When the menu is active, the MIDlet screens behind it are not updated (paint events are not sent beyond the menu). As a result, background threads could still be updating internal states, but the application appears to be paused.

To solve this problem, you can put the code that updates the application into the paint method, and then have the main application thread invoke repaint() in a continuous loop. When the menu is closed, repaint() invokes paint() to update the screen and the state. When the menu is open, paint() is not invoked, so the application is effectively paused.
# Using key mappings for games

The BlackBerry Wireless Handheld supports the following default key mappings for MIDlet games.

<table>
<thead>
<tr>
<th>Keys</th>
<th>Meaning</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>R D F G</td>
<td>UP LEFT DOWN RIGHT</td>
<td>These keys are useful for playing games with one hand.</td>
</tr>
<tr>
<td>-OR-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U H J K</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R D G C</td>
<td>UP LEFT RIGHT DOWN</td>
<td>These keys are useful for playing games with both hands.</td>
</tr>
<tr>
<td>-OR-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U H K N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U A S J</td>
<td>UP LEFT RIGHT DOWN</td>
<td>These keys are useful for playing games with both hands. Users do not have to take their thumbs off the keys.</td>
</tr>
<tr>
<td>R F L &lt;</td>
<td>UP DOWN LEFT RIGHT</td>
<td></td>
</tr>
</tbody>
</table>
2: Designing MIDlets
Comparing MIDP and BlackBerry UI APIs

This section provides information on the following topics:

- BlackBerry and MIDP APIs
- Comparison of UI components
BlackBerry and MIDP APIs

When you write an application for BlackBerry handhelds, you can use one of two API sets for user interfaces:

- MIDP UI APIs (javax.microedition.lcdui package)
- BlackBerry UI APIs (net.rim.device.api.ui packages)

If you are writing an application to run on any MIDP-compliant device, use the MIDP UI APIs. If you are writing an application specifically for BlackBerry handhelds, use the BlackBerry UI APIs. BlackBerry APIs provide access to specific features of the handheld and enable more sophisticated UI layout and interaction.

⚠️ Note: The BlackBerry UI APIs and MIDP UI APIs are mutually exclusive. You cannot use both API sets in the same application.

This section compares the MIDP UI APIs with BlackBerry UI APIs. Refer to the BlackBerry Java Developer Guide Volume 1 – Fundamentals for more information on using BlackBerry UI APIs.

Using screens

The BlackBerry UI structures components differently from the MIDP UI. MIDP separates UI components into high, middle, and low-level APIs. Certain Displayable objects in MIDP (such as List and Alert) are considered field components in the BlackBerry UI. Because you can only aggregate MIDP items to Form displayables, you cannot combine List and Alert objects with other components in MIDP. With the BlackBerry UI API, you can combine any field object with others.

Using the MIDP UI components, you cannot display more than one Displayable object on the screen at a time. For example, you cannot combine a List with another type of UI component, such as a Gauge. The BlackBerry UI, however, does not classify components by level, nor does it restrict you to only a certain combination of components for each screen. You can combine any number of field objects together into a Manager, such as a ListField, GaugeField, or numerous BitmapField objects. In this respect, the BlackBerry UI is more extensible, which enables you to create complex application interfaces more efficiently.
Layout managers

The BlackBerry API provides layout managers that enable you to display fields vertically, horizontally, or both. You declare a layout manager at the top level of your screen, and then add field components to it.

The closest relation to a layout manager in the MIDP API is the Form component. It aggregates Item objects vertically on the display.

When you decide whether to use the BlackBerry APIs or the MIDP APIs, you should carefully consider how you intend to organize content on the interface. If your interface involves frequent scrolling between fields, or a horizontal field layout, you might want to use the layout managers of the BlackBerry API.

Alerts

The BlackBerry UI has two main types of Screen objects: PopupScreen, which contains Dialog and Status, and FullScreen, which contains MainScreen. A PopupScreen is not treated as a specific Screen object; you are not required to push it onto the context stack.

Because MIDP is intended for devices with minimal displays, such as mobile phones, an Alert object functions differently from PopupScreen in the BlackBerry API. Alert objects are screens that use the entire display when they are invoked.

The BlackBerry UI enables you to display information that is supplementary to your application in a pop-up screen without interrupting the “flow” of the application. Users do not have to leave the main screen, and then return to it later.

Using commands

MIDP devices typically display commands as soft keys (buttons on the screen), whereas the BlackBerry handheld displays commands as menu items.

Menus

The MIDP UI prioritizes commands according to the device specifications and assigns certain commands to soft keys. BlackBerry handholds organize commands on a menu that is accessible by clicking the trackwheel. You can specify both the order of menu items and the menu item that is selected by default when the menu opens.
Comparing MIDP and BlackBerry UI APIs

Commands
Using the MIDP UI components, you can assign commands directly to Displayable objects. The BlackBerry UI, however, contains a Menu object. You add commands to the application's Menu object, which can be applied to the entire application. Menu items can also be specific to a particular screen in the application.

You can tailor specific menu items to specific parts of your application by creating context menu items.

Listeners
The MIDP UI includes one default type of listener, CommandListener. You can only register one CommandListener for each Displayable object. After you register a CommandListener with the Displayable object, you can write code to perform an action when that object is selected. For example, you can assign a command listener to a Form, and then perform an action when the user selects an item in the Form. The MIDP UI also contains functions for basic event handling that are based on keyboard input.

The BlackBerry UI offers a more flexible implementation of listeners, based on the more involved level of components that can be structured on a screen. The BlackBerry UI contains a generic event listener model, which enables you to implement listeners for many types of events, such as a user pressing a key or clicking the trackwheel, or field, focus, and scrolling changes.

Comparison of UI components
The following table compares the BlackBerry UI components to the MIDP UI.

<table>
<thead>
<tr>
<th>BlackBerry</th>
<th>MIDP</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoTextEditField</td>
<td>-</td>
<td>AutoTextEngine is specific to the BlackBerry handheld.</td>
</tr>
<tr>
<td>BasicEditField</td>
<td>TextField</td>
<td>TextField is an editable text field.</td>
</tr>
<tr>
<td>BitmapField</td>
<td>Image</td>
<td>The BlackBerry UI BitmapField objects are similar to the Image object in the MIDP UI. Bitmap is equivalent to ImageItem.</td>
</tr>
<tr>
<td>ButtonField</td>
<td>-</td>
<td>You can create a ButtonField equivalent in MIDP by using a variety of drawing and focusing methods on a Canvas.</td>
</tr>
<tr>
<td>CheckboxField</td>
<td>ChoiceGroup</td>
<td>The MIDP equivalent of a CheckboxField is a ChoiceGroup with multiple selection specified.</td>
</tr>
<tr>
<td>ChoiceField</td>
<td>-</td>
<td>The MIDP ChoiceGroup Item is similar to a RadioButtonField or CheckboxField.</td>
</tr>
<tr>
<td>DateField</td>
<td>DateField</td>
<td>DateField objects are editable in both the BlackBerry and MIDP APIs.</td>
</tr>
<tr>
<td>Dialog</td>
<td>-</td>
<td>MIDP does not support pop-up screen functionality. The Alert displayable serves as the MIDP pop-up screen component.</td>
</tr>
</tbody>
</table>
Comparison of UI components

<table>
<thead>
<tr>
<th>BlackBerry</th>
<th>MIDP</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>EditField</td>
<td>-</td>
<td>EditField is an extension of BasicEditField that enables users to type special characters in the field.</td>
</tr>
<tr>
<td>GaugeField</td>
<td>Gauge</td>
<td>A MIDP Gauge must be declared as interactive or noninteractive. For a GaugeField to be interactive, it must be created with the Field.EDITABLE style.</td>
</tr>
<tr>
<td>LabelField</td>
<td>StringItem</td>
<td>A non-editable text field.</td>
</tr>
<tr>
<td>ListField</td>
<td>List (Displayable)</td>
<td>A List in MIDP provides a list of options, which can be used instead of a menu. Since List is a Displayable object, you cannot add other UI components to the screen at the same time.</td>
</tr>
<tr>
<td>Menu</td>
<td>-</td>
<td>Commands in a MIDP application appear as menu items on the BlackBerry handheld.</td>
</tr>
<tr>
<td>NullField</td>
<td>-</td>
<td>A NullField is a field that has no size. It is often used as a source for a focus change listener.</td>
</tr>
<tr>
<td>NumericChoiceField</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ObjectChoiceField</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ObjectListField</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PasswordEditField</td>
<td>TextField</td>
<td>In MIDP, a PASSWORD constraint constant can be added to a TextField; characters are not echoed on the display.</td>
</tr>
<tr>
<td>RadioButtonField</td>
<td>ChoiceField</td>
<td>The MIDP equivalent of a RadioButtonField is a ChoiceField with explicit selection specified.</td>
</tr>
<tr>
<td>RadioButtonGroup</td>
<td>-</td>
<td>Options in a MIDP ChoiceField are grouped by default.</td>
</tr>
<tr>
<td>RichTextField</td>
<td>-</td>
<td>In MIDP, style attributes (such as STYLE_ITALIC and STYLE_BOLD) can be applied to text drawn onto a Canvas.</td>
</tr>
<tr>
<td>SeparatorField</td>
<td>-</td>
<td>The separator is a user interface component that is specific to standard BlackBerry application interfaces.</td>
</tr>
<tr>
<td>Status</td>
<td>Alert</td>
<td>An Alert can be modal, which requires user interaction, or timed, is closes automatically. An Alert is not a pop-up screen, but serves a similar purpose.</td>
</tr>
<tr>
<td>-</td>
<td>Ticker</td>
<td>The BlackBerry UI API does not contain an equivalent for the MIDP Ticker item.</td>
</tr>
<tr>
<td>TreeField</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
3: Comparing MIDP and BlackBerry UI APIs