

MIDP Command Mappings In Nokia Series 40 Devices

Version 1.0; May 18, 2004

Java™

NOKIA

Copyright © 2004 Nokia Corporation. All rights reserved.

Nokia and Nokia Connecting People are registered trademarks of Nokia Corporation. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. Other product and company names mentioned herein may be trademarks or trade names of their respective owners.

Disclaimer

The information in this document is provided “as is,” with no warranties whatsoever, including any warranty of merchantability, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. Furthermore, information provided in this document is preliminary, and may be changed substantially prior to final release. This document is provided for informational purposes only.

Nokia Corporation disclaims all liability, including liability for infringement of any proprietary rights, relating to implementation of information presented in this document. Nokia Corporation does not warrant or represent that such use will not infringe such rights.

Nokia Corporation retains the right to make changes to this specification at any time, without notice.

License

A license is hereby granted to download and print a copy of this specification for personal use only. No other license to any other intellectual property rights is granted herein.

Contents

1	Overview	5
2	Detailed Description.....	6
2.1	General Rules for Command Mapping	8
2.2	Middle Softkey (MSK)	8
2.3	Right Softkey (RSK).....	9
2.4	Left Softkey (LSK)	9
2.5	The Order of Commands in Command Lists	9
2.6	Item Commands.....	9
2.7	Default Item Command	10
3	Examples	11
3.1	Example Code	11
3.2	Two Softkeys and a Normal Keypad	12
3.3	Two Softkeys and a Foldout Keyboard.....	12
3.4	Three Softkeys and a Normal Keyboard.....	13
3.5	Three Softkeys and a Foldout Keyboard	14
3.6	Softkeys in Full Screen Mode Canvas in MIDP 2.0	15
4	Summary	17
5	References	18
Appendix A Common Softkey Mappings in Series 40 Developer Platform 2.0 Devices with Three Softkeys		19

Change History

May 18, 2004	Version 1.0	Initial document release

1 Overview

Commands are user interface objects that are used to transmit user interaction. They are mapped to the device's softkeys depending on the type and priority of the command. This enables Java™ MIDP applications (MIDlets) easily to adhere to device look and feel while remaining independent of the device.

Nokia Series 40 devices have either two or three softkeys. The softkeys are located in the left and right bottom corners of the display and the possible third softkey is located in the middle. Because the amount of the softkeys differs, there are also differences in command mappings.

In addition, some Nokia Series 40 devices (for example, Nokia 6810 and Nokia 6820) have a foldout keyboard, which has an effect on command mapping. A foldout keyboard means that normally the ITU-T9 keypad is in use, but the keypad can be opened to access a full qwerty keyboard. Opening the keyboard also rotates the display 90 degrees to the right or the left, which again affects softkey mapping and must be taken into account in MIDlet usability design.

These issues may cause developers extra challenges. The purpose of this document is to describe how commands are mapped in Nokia Series 40 devices that are currently available. With the help of this document, the developer can easily see how the same commands are mapped in different Nokia Series 40 devices.

If you need more information about commands, please refer, for example, to the document *MIDP 1.0: Introduction to MIDlet Programming* available through www.forum.nokia.com.

2 Detailed Description

As specified in the *MIDP Specification*, a MIDlet can associate Command objects with Displayable objects. In case of Nokia devices, Command objects are directly mapped to softkeys or a softkey's Options menu. This applies to Screen and Normal Canvas displays but not to the Nokia UI API's FullCanvas displays. Commands cannot be added to FullCanvas displays at all. However, it is possible to get normal key events from softkeys in FullCanvas. In Figure 1, the OK and Exit commands have been added to a Form.



Figure 1: Commands mapped to the softkeys

The left side of Figure 2 presents the very same Form after adding a **Help** command to it. In this case, the implementation changes the left softkey label to **Options**. The right side of Figure 2 presents the command list that appears when the user presses the **Options** key. There are two commands in the command list: **OK** and **Help**. The commands **Select** and **Back** are added to the command list automatically by the implementation.

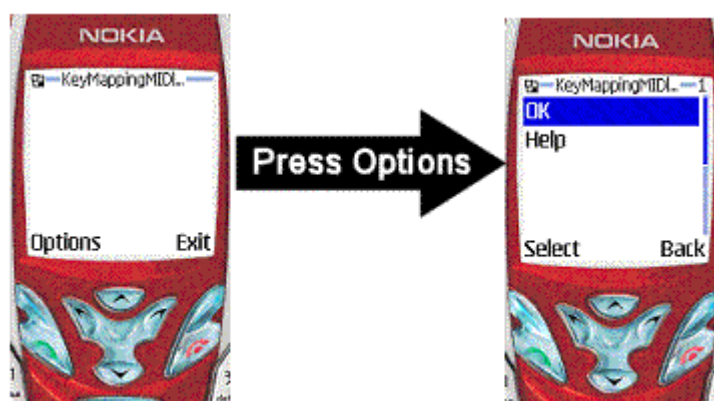


Figure 2: Two commands mapped to the left softkey

In MIDP 2.0 Full Screen Canvas displays, Command objects are mapped to the softkeys but the softkey labels are not displayed. If a CommandListener has been added to the Canvas and the user presses a softkey, an options menu containing all commands in the order defined by the command priority rules is displayed. The command priority rules are explained in Section 2.1, “General Rules for Command Mapping.” If there is no CommandListener, pressing any softkey will deliver low-level key events to the MIDlet as in the Nokia FullCanvas display. Those low-level key events can be caught by using the `keyPressed()` or `keyRepeated()` methods of the Canvas class. Pressing softkeys returns negative values: LSK = -6, RSK = -7, and MSK = -5.

Each command has the following attributes:

1. **Label:** the text string to be shown to the user, representing the command on the display. If the label text is too long to fit on the assigned softkey or choice item, it is truncated with an ellipsis. Otherwise, labels specified by the MIDlet will be used as supplied and will not be modified by the implementation. A command can have two labels in MIDP 2.0:
 - A short label: This will be used as the text for the command if it is mapped to a softkey. If the short label is too long, it is truncated with an ellipsis.
 - A long label: This will be used if the command is displayed as a choice item in a command list. The short label is used if the MIDlet does not define a long label or if the long label is too long.
2. **Type:** an indication of the command's intent.
3. **Priority:** an integer value that the MIDlet can use to describe the command's priority when compared to other commands of the same type, where the command with the lowest integer value has the highest priority.

The command type fields defined in the *MIDP Specifications* are summarized in Table 1. The "Intent of command" column indicates the normal interpretation of the semantics of the command information given, however, there is no automatic implementation and the action to be taken is defined by the application's `CommandListener`.

The table also explains to which softkey the Command would be mapped to by default.

- RSK = Right Softkey
- LSK = Left Softkey
- MSK = Middle Softkey (if there are three softkeys)

The Nokia default placement of the softkeys follows the rule that the RSK should only have negative type commands (`STOP`, `CANCEL`, `BACK`, and `EXIT`) mapped to it. Positive type commands (`OK`, `ITEM`, `SCREEN`, and `HELP`) should always be mapped to the MSK or the LSK.

It is important to use correct command types to adhere to the correct user interface style. In other words, if a command with a negative label has a positive command type parameter, the command has been mapped to an incorrect softkey.

Default softkey labels are used if the MIDlet assigns an empty label or a string containing only white space. An exception is thrown if the MIDlet assigns a null label. The commands in the following table are listed in the command type priority order.

MIDP command type	Intent of command	Nokia default placement	Nokia default command label
STOP	Stop currently running process or operation.	RSK	Stop
OK	Standard positive response to a dialogue in the current display.	MSK / LSK	OK
CANCEL	Standard negative response to a dialogue in the current display.	RSK	Cancel
ITEM	MIDlet-defined command relevant to a specific item in the display. Context	MSK / LSK	Select

	sensitive.		
SCREEN	MIDlet-defined command relevant to the current display.	MSK / LSK	Select
HELP	Request for MIDlet help information.	MSK / LSK	Help
BACK	Navigation command which returns the user to the logically previous display.	RSK	Back
EXIT	Exit the MIDlet.	RSK	Exit

Table 1: MIDlet commands

2.1 General Rules for Command Mapping

If there is more than one command with the highest priority type, the following rules are applied to determine which command has the highest priority.

1. If there is more than one command of the same type, then the command priority is based on the command priority value specified by the command object itself.
2. If, after applying rule 1, there are still commands with the same priorities (for example, more than one command with the same type and priority), their order is based on the order the commands were programmatically added to the displayable (“first come, first served”).

Note that UI operations provided by the implementation might override the priority rules for the command placements on softkeys. For example, the implementation adds the **Edit** command to MSK or LSK when a **TextField** object is focused (see Figure 3).



Figure 3: UI provided operation added to the MSK by the implementation

2.2 Middle Softkey (MSK)

If the device has an MSK, the positive command with the highest priority is mapped directly to it and the other positive commands are placed in the **Options** menu of the LSK as command list entries. Negative commands are never mapped to the MSK, but they are mapped either to the RSK or into an options list on the LSK. Determining which command is mapped directly to the MSK is done using the following priorities based on type (highest priority first): **OK**, **ITEM**, **SCREEN**, and **HELP**.

2.3 Right Softkey (RSK)

If there are two or more RSK commands, the command with the highest priority is mapped to the right soft key, while the other(s) is/are mapped to the **Options** menu of the LSK as command list entries. Positive commands are never mapped to the RSK. Determining which RSK command is mapped directly to the right softkey is done using the following priorities based on type (highest priority first): **STOP**, **CANCEL**, **BACK**, and **EXIT**.

2.4 Left Softkey (LSK)

If there are two or more commands that are not mapped directly to the MSK or RSK, these are all mapped to the **Options** menu of the left softkey as command list entries.

If the device has only two softkeys and only one command is to be mapped to the LSK, the Command is mapped directly to the LSK.

If a device has three softkeys, the highest priority positive command will be mapped to the MSK and any other positive commands will be displayed in an Options list on the LSK as a command list entry. This is true even if there is only a single additional command. The LSK on a three-softkey device will never be directly mapped to a command; it always appears as an Options list.

2.5 The Order of Commands in Command Lists

Commands within a command list are displayed in the following order (the first one is the topmost):

1. Any UI component operation (native software UI component operations, for example, internal selection element in ChoiceGroup).
2. Default Item command (for further details, see Section 2.7, “Default Item Command”).
3. Item commands (for further details, see Section 2.6, “Item Commands”).
4. Commands by command type order (**STOP**, **OK**, **CANCEL**, **ITEM**, **SCREEN**, **HELP**, **BACK**, and **EXIT**).

If there is more than one of any type of command within each command type group placement, as shown above, individual commands are ordered according to the rules that are presented in Section 2.1, “General Rules for Command Mapping.”

2.6 Item Commands

Note that this section handles only the MIDP 2.0 feature that enables adding commands to Item objects.

It is possible to add commands to Item within a Form in MIDP 2.0. These commands should not be confused with commands with type **ITEM** in general. Item commands are context sensitive commands for the specific Items and are available either through the softkeys or the Options menu.

When Item commands are presented in the Options menu that also contains Commands set to the Form containing the Item, the Item Commands have a higher priority and therefore they are presented before the Form Commands (and before Form **ITEM** type Commands). See Section 2.5, “The Order of Commands in Command Lists” for further details. The ordering of Item commands is based completely on priority and the command type is ignored.

If an Item has got a default command, it will have a higher priority than Item commands. See Section 2.7, “Default Item Command” for further details.

In devices that have two softkeys, the Item command with the highest priority is placed on the left softkey if there are no additional commands, or topmost in the Options menu if there are other commands as well.

In devices that have three softkeys, the Item command with the highest priority is placed on the MSK.

2.7 Default Item Command

Items have an assumption of the default command, and always give it the highest priority. If there is no default Item command, the highest priority returns to the Item commands as described in the previous section.

Inline editing is not enabled within the Form and as a result, the default Item Command will not have the highest priority in Gauge, TextField, and DateField, whereas the UI component operation Edit has the highest priority.

Regardless of whether a Choice Group has a default Item command and/or Item commands, the highest priority command is always the internal element selection, that is, the Select operation in EXCLUSIVE, Mark/Unmark operations in MULTIPLE, and Open in POPUP.

The Send key can also be used to execute the default Item command in all Items with the exception of ChoiceGroup, where the Send key is mapped to the internal element selection command.

3 Examples

This chapter presents a simple example MIDlet that includes a Form, which again contains a couple of commands and a TextField. The MIDlet is used to illustrate differences between command mappings in the following situations:

- Two softkeys – normal keypad
- Two softkeys – foldout keyboard
- Three softkeys – normal keypad
- Three softkeys – foldout keyboard

These differences are presented in the following sections.

Two command types have been added to the Form: **OK** and **EXIT**. In addition, there is one default Item Command (**Edit**) for TextField, which is added by the implementation. As defined before, the priority order of those commands is:

1. **Default Item Command**
2. **OK**
3. **EXIT**

3.1 Example Code

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class KeyMappingMIDlet
    extends MIDlet
    implements CommandListener {

    protected Command exitCommand = new Command("Exit", Command.EXIT, 1);
    protected Command okCommand = new Command("OK", Command.OK, 1);
    protected Form form;
    protected TextField textField;

    public KeyMappingMIDlet() {
        form = new Form("KeyMappingMIDlet");
        textField = new TextField("Text Field", null, 30, TextField.ANY);
        form.append(textField);
        form.addCommand(okCommand);
        form.addCommand(exitCommand);
    }

    protected void pauseApp() {
    }

    protected void startApp() throws javax.microedition.midlet.
        MIDletStateChangeException {
        form.setCommandListener(this);
        Display.getDisplay(this).setCurrent(form);
    }

    protected void destroyApp(boolean parm1) throws
        javax.microedition.midlet.
        MIDletStateChangeException {}

    public void commandAction(Command cmd, Displayable disp) {
        if (disp == form) {
            if (cmd == exitCommand) {
                try {

```




Figure 5: Two softkeys and a foldout keyboard

This category includes, for example, Nokia 6800.

3.4 Three Softkeys and a Normal Keyboard

Three softkeys with a normal keypad is getting more common, because all new Series 40 device models have three softkeys. Figure 6 illustrates how the example MIDlet appears in these kind of devices.

Note that the TextField Item's default command **Edit** is shown in the middle softkey when the TextField is focused on the Form. Also note that the **Select** and **Back** commands are added to the command list by the implementation, and that the **Select** command is placed on the middle softkey.

For a list of common softkey mappings, refer to Appendix A, "Common Softkey Mappings in Series 40 Developer Platform 2.0 Devices with Three Softkeys."



Figure 6: Three softkeys and a normal keypad

This category includes, for example, Nokia 5140, Nokia 6230, Nokia 6650, and Nokia 7600.

3.5 Three Softkeys and a Foldout Keyboard

This situation is exactly as with three softkeys on a normal keyboard when the normal keypad is used, that is, the full keyboard is closed. However, there are differences when the full keyboard is opened. When the full keyboard is used, the display is rotated 90 degrees left or right and the commands on the left and right softkeys are mapped to different keys, but they are still in the left and right bottom corners of the display and work as when the keyboard is closed. The biggest difference is that the command in the middle softkey is mapped to the bottom middle of the display, although the key itself is on the right side of the display (see Figure 6). In practice, the middle softkey is not changed as the left and right softkeys are, but only the location of the label is changed.

For a list of common softkey mappings, refer to Appendix A, “Common Softkey Mappings in Series 40 Developer Platform 2.0 Devices with Three Softkeys.”



Figure 7: Three softkeys and a foldout keyboard

This category includes, for example, Nokia 6810 and Nokia 6820.

3.6 Softkeys in Full Screen Mode Canvas in MIDP 2.0

In Full Screen Canvas displays based on MIDP 2.0, Command objects are mapped to the softkeys but softkey labels are not displayed. Figure 8 shows an example where **EXIT** and **OK** commands are added to the GameCanvas display that is set to the Full Screen mode. When the user presses one of the softkeys, the Command menu list is displayed and all the commands are listed in the command list in the command priority order.

Note that the functionality of some Nokia Concept SDK/emulators (for example, Nokia 6230 MIDP Concept SDK Beta 0.2) may be incorrect.



Figure 8: Softkeys in Full Screen mode GameCanvas in MIDP 2.0

In the normal mode both MIDP 1.0 and MIDP 2.0 Command objects are displayed as in other Displayable components (see Figure 9). The positive command that has the highest priority is mapped to the middle softkey and the other positive commands are mapped to the left softkey. The negative command having the highest priority is mapped to the right softkey and the other negative commands are mapped to the command list menu of the left softkey.



Figure 9: Softkeys in normal mode Canvas

4 Summary

This document presented how commands are mapped in Nokia Series 40 devices. Nokia Series 40 devices have either two or three softkeys, and this has an effect on command mapping. Command mapping is based on the nature of the command (positive or negative commands), their priority, and the order that they are programmatically added to a specific UI component.

Positive commands are always mapped either to the middle or left softkey and negative commands are always mapped to the right softkey or the Option menu under the left softkey. Both the middle and the right softkeys can include only one command and the other commands are mapped to the command menu list under the left softkey in the priority order. It is important to use positive Command Type parameters for positive commands and negative Command Type parameters for negative commands to adhere to the correct user interface style.

A foldout keyboard alters command mapping especially when the device has three softkeys.

5 References

MIDP 1.0: Introduction to MIDlet Programming. Available through <http://www.forum.nokia.com/>

MIDP 1.0 Specification (JSR 37). Available through <http://java.sun.com/products/midp/>

MIDP 2.0 Specification (JSR 118). Available through <http://java.sun.com/products/midp/>

Appendix A Common Softkey Mappings in Series 40 Developer Platform 2.0 Devices with Three Softkeys

The middle softkey (MSK) displays the primary function of the view and the left softkey (LSK) gives an options list, if applicable. The right softkey (RSK) displays the negative actions (for example, **Exit**, **Back**, and **Clear**).

The following rules apply to softkeys:

- When a softkey provides a function, it must be labeled.
- The same function cannot be available in two softkeys at the same time.
- A certain softkey function has to be mapped to the same softkey in all contexts.

Some common default labels of softkeys are listed in Table 2. Note that these labels are not Java-specific but apply to Series 40 applications in general.

View	Softkey label		
	LSK	MSK	RSK
Idle state	Go to	Menu	Names or user-defined
Exclusive selection list (e.g., Menu)	[Options] Used if applicable.	Select Used in menus and option lists.	Exit*/Back*
Multiple selection list	Options Used in menus, if applicable.	Mark/Unmark	Back/Done Back is used when no changes have been made. Done is used when changes have been made.
Confirmation queried by the system	Yes Confirms a query.	[empty]	No Discards the query.

View	Softkey label		
	LSK	MSK	RSK
Editors (text boxes)	<p>Options</p> <p>Used when both fetch services are available and it is necessary to allow an empty user entry. The Options list includes Search.</p> <p>Also used when there are T9 or other editor functions available in the query editor.</p>	<p>OK/Search</p> <p>OK is the default label in data queries and editors.</p> <p>Search is used in empty data queries, if the fetch service is available and an empty user entry is not allowed. Note that in Java applications Search appears in the Options list of a TextBox if the type of the TextBox is PHONENUMBER or EMAILADDR.</p>	<p>Back/Clear</p> <p>Back is used when no characters have been entered to the data query.</p> <p>Changed to Clear when characters have been entered.</p>
Viewers (e.g., Gallery)	<p>Options</p> <p>Used in menus, if applicable.</p>	[application specific, e.g., Zoom or Play]	Exit*/Back*
Interruption	<p>Options</p> <p>Used in menus, if applicable.</p>	[application specific]	<p>Quit</p> <p>Interrupts an ongoing operation.</p> <p>Used when restoring the original state cannot be guaranteed.</p>
Stopping	<p>Options/Stop (Stop is used only in alarm clock)</p> <p>Options is used in menus, if applicable.</p>	[application specific]	<p>Stop</p> <p>Can be used instead of Quit, if a familiar application using the Stop label is integrated in the device.</p>
Cancellation	<p>(Options)</p> <p>Options is used in menus, if applicable.</p>	[application specific]	<p>Cancel</p> <p>Pressing Cancel truly cancels an ongoing operation. Used when no permanent changes have been made yet and the original state can be restored.</p>
Soft notifications	<p>Options</p> <p>Used in menus, if applicable.</p>	[application specific]	Exit*

Table 2: Default softkey labels

* The **Exit** label is used as the RSK text, when pressing it returns the phone to the menu.

* The **Back** label is used as the RSK text, when pressing it returns the phone to the previous view.