FORUM NOKIA

Series 40 UI Style Guide

Version 3.3; October 2, 2007

Series 40

NOKIA

Copyright © 2004-2007 Nokia Corporation. All rights reserved.

Nokia and Forum Nokia 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. 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.

The phone UI images shown in this document are for illustrative purposes and do not represent any real device.

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 Introduction 7
1.1 Purpose 7
1.2 Audience 7
2 Series 40 UI style overview 8
2.1 Principles of Nokia UI design 9
3 Displays 10
3.1 Basic layouts 10
3.1.1 Idle layout 10
3.1.2 Base layout with a scroll bar 11
3.1.3 Base layout without a scroll bar 12
3.1.4 Browser layout 12
3.1.5 Java application layouts 13
3.2 Layout design guidelines 16
3.2.1 Icon sizes 17
3.3 Fonts 18
3.4 Presentation of text 20
3.4.1 Marquee scrolling 20
3.5 Graphical elements 21
4 Keys 23
4.1 Key press 23
4.2 Softkeys 24
4.3 Scroll keys 25
4.4 Send key 26
4.5 End key 26
4.6 Numeric keys 26
4.7 Power key 26
4.8 Volume keys 27
4.9 Keyguard 27
5 Interaction 28
5.1 Idle state 28
5.1.1 Idle personalization 29
5.2 Menu 29
5.3 Navigation 30
5.3.1 Four-way navigation 30
5.4 Navigation aids 32
5.4.1 Focus 32

5.4.2 Header text ... 32
5.4.3 Scroll bar ... 32
5.4.4 Index numbers ... 33
5.5 Backstepping ... 33
5.6 Text input ... 34
5.7 Feedback ... 35
6 Sounds ... 36
7 Localization ... 37
8 UI components ... 38
8.1 List types ... 39
8.1.1 One-row choice items ... 39
8.1.2 Multiple-row choice items ... 39
8.2 Settings ... 40
8.3 Multiselection lists ... 40
8.4 Option lists ... 41
8.5 Grids ... 41
8.6 Tabs ... 42
8.7 Data queries ... 44
8.8 Forms ... 44
8.9 Confirmation and list queries ... 46
8.10 Soft notifications ... 47
8.11 Notes ... 48
9 Terms and abbreviations ... 49
10 Evaluate this resource ... 52

Figures

Figure 1: Display resolution and softkey combinations supported in the history of the Series 40 UI. 8
Figure 4: Idle state 28
Figure 5: Elements shown in the idle state 28
Figure 6: Series 40 active standby – left display is inactive mode, right display shows active mode 29
Figure 7: Series 40 UI menu style examples – grid, list, and tab menu (tab only in 240 x 320 resolution displays) 30
Figure 8: Example of a basic navigation hierarchy 30
Figure 9: Example of scrolling a grid with a four-way scroll 31
Figure 10: Navigation aids in the Series 40 UI 32
Figure 11: Examples of text input screens 34
Figure 12: Example of layered pop-ups 38
Figure 13: Example of one-row choice items 39
Figure 14: Example of multiple-row choice items 39
Figure 15: Example of a settings screen 40
Figure 16: Example of a multiselection list 40
Figure 17: Examples of a grid 41
Figure 18: Example of tabs 42
Figure 19: Examples of data query screens 44
Figure 20: Examples of forms 44
Figure 21: Examples of confirmation and list queries 46
Figure 22: Example of a soft notification 47
Figure 23: Soft notification with a list (combining communication events that have occurred in user's absence) 47
Figure 24: Example of a note 48

Change history

September 7, 2004	Version 2.0	Replaces the Forum Nokia document *Nokia Series 40 UI Style Guide v1.0*.
March 10, 2005	Version 2.1	Information about 208 x 208 resolution display added.
December 21, 2005	Version 3.0	Information about Series 40 3rd Edition added.
November 30, 2006	Version 3.1	Updates throughout the document to include Series 40 3rd Edition, Feature Pack 2 features. Information about font size change in 240 x 320 resolution display added. Information specific to older display resolutions and key combinations removed.
May 4, 2007	Version 3.2	Updates about the Series 40 UI style to reflect Series 40 5th Edition changes. Main changes include: list query UI component, marquee scrolling of the text and scroll left/right shortcuts in form selection fields.
October 2, 2007	Version 3.3	Updates include new features of Series 40 5th Edition, Feature Pack 1: tabs (new UI component), soft notification component with a list.

1 Introduction

1.1 Purpose

This document provides an overview of the Nokia Series 40 User Interface (UI) and describes its essential parts. It can be used as an introduction to the Series 40 UI or for reference. It can also be used as background material to help with decision-making in different UI design phases.

The style guide is intended to be a compact, easy-to-read guide, concentrating on the Series 40 UI in general, with enough guidelines to help developers design good applications. However, specific details (for example, how to optimize Java™ MIDlets, browser content for Series 40 devices, and introduction to Macromedia Flash Lite from Adobe) are not included, as that information is available in other documents. For more information, see the following Forum Nokia (http://www.forum.nokia.com) documents:

- *Game User Experience Library*
- *MIDP Command Mappings in Nokia Series 40 Devices*
- *Series 40 Platform: Designing XHTML Mobile Profile Content*
- *Using Icons in MIDlets (with Example)*
- *Introduction to Flash Lite*

The style guide gives detailed information about supported display resolutions and key combinations. Information specific to older display resolutions and key combinations can be found in earlier versions of this document. The following items, for example, are no longer included:

- Information specific to 96 x 65 resolution display;
- Information specific to 128 x 128 resolution display;
- Information specific to 208 x 208 resolution display;
- Information specific to two softkeys;
- Information specific to two-way navigation;
- Information specific to full alphanumeric keyboard (QWERTY).

1.2 Audience

This document is intended for anyone who needs a general overview of Series 40 UI styles. It is specifically targeted at:

- Application developers
- Content providers
- UI interaction designers
- UI software engineers
- UI graphic designers
- Usability experts

2 Series 40 UI style overview

The Series 40 UI is a highly flexible and configurable UI platform — easy to use and supportive of a variety of different UI styles and feature configurations. For example, the following display resolutions are supported in the Series 40 UI.

Display resolutions

128 x 160	240 x 320
	

Table 1: Display resolutions

The Series 40 UI is intended for one-hand operation, meaning that the user can perform tasks with one hand. Pressing two keys simultaneously is not recommended.

The Series 40 UI represents an evolution of the two-softkey UI introduced by Nokia in 1994; Nokia published the first Series 40 products in 2001. Products such as the Nokia 8310 device and the Nokia 6310i device were among the first devices implemented on top of the Series 40 platform, the latter being the first one with Java™ platform support. The actual Series 40 platform was introduced together with the Nokia 7210 device.

Over the past years, the Series 40 UI has evolved from a small-resolution, black-and-white UI to a high-resolution color UI with an enhanced graphical look and new UI components such as forms. With the increasing number of supported features, the more advanced three-softkey UI logic has become the standard, rather than the two-softkey UI logic.

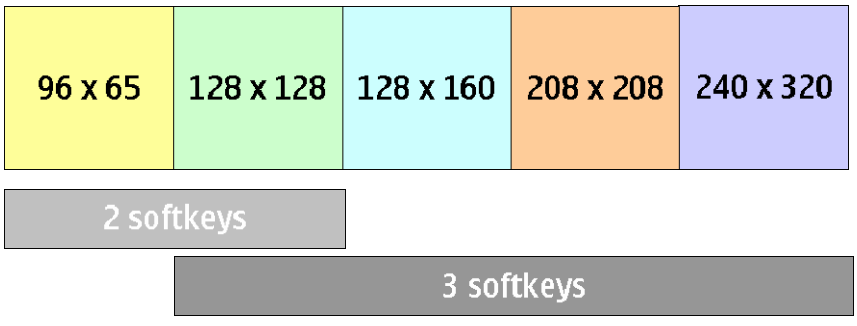


Figure 1: Display resolution and softkey combinations supported in the history of the Series 40 UI

2.1 Principles of Nokia UI design

The following three design principles form the basis for a Nokia UI.

Simplicity

A Nokia user interface is simple and elegant.

Interaction is smooth; there are no complicated or unnecessary elements. The user can find all needed functions easily and perform tasks effectively, without getting tangled in secondary issues.

Graphics and display layouts are uncluttered, crisp, and simple. Displays show the most important information clearly, rather than squeezing in as much data as possible. Text fonts are clear and comfortable to read. There is no excess decoration.

Sounds are clear and pleasing to the ear. Each sound has a function. There are no excess sounds.

The interaction, display text, graphics, and sound design are consistent and harmonious. A single element always appears in the same format and behaves in the same way. All elements are in balance and support each other.

Classic quality is achieved by paying attention to details.

Friendliness

A friendly UI feels familiar and is intuitive to use. It is based on elements the user already knows, so there is no need to learn complicated procedures.

The UI has a polite tone; it does not command the user or use offensive words. The user can trust the UI; it does not mislead.

The UI feels personal to each user. It is localized and uses the language of the user and symbols the user recognizes. It can be personalized to fulfill the needs and desires of each user.

The product is attractive and fun to use. It has elements that give pleasure to the user. The user experience is delightful and invites people to work and play with the device.

Power

The UI permits effective access to the applications in the device. The user can accomplish tasks quickly, in a straightforward manner, and without errors, using the provided applications.

Applications and features that are used frequently can be quickly and easily located within the UI. The UI is flexible, allowing tasks to be accomplished in different ways, depending on individual habits and desires. Advanced users are offered shortcuts for even faster access to often-needed functions.

The interface is responsive, giving instant feedback to the user. The user feels in control.

3 Displays

The Series 40 UI supports display resolutions of 128 x 160 and 240 x 320 (width x height) pixels.

The display is divided into the following generic components:

- Status zone: Displays the header text, index numbers, and status indicators.
- Main zone: Displays the content of the current application, for example, user-selectable items, user data, and message contents.
- Softkey zone: Displays the softkey labels. Divided into three parts depending on the number of softkeys.

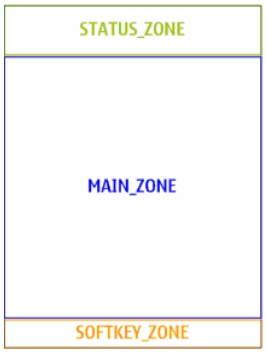


Figure 2: Basic layout

A minimum of 4,096 colors is preferred for color displays.

The display supports the availability of status information throughout the UI. There are separate zones for status indicators and header text, which enables global status indicators and signal and battery bars to be shown in every state of the device (except when an application is used in full screen mode, if supported).

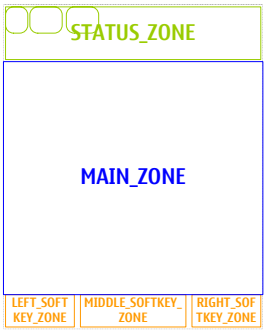
3.1 Basic layouts

The following sections define the main zone and softkey zone sizes for the basic Series 40 layouts. The examples represent layouts in left-to-right languages. In right-to-left languages, the layouts are mirrored (some elements, such as the softkey zones and signal and battery bar zones, still keep the same position as in the layouts for left-to-right languages).

Note: The minimum font size in the Khmer language in a 128 x 160 resolution display is 16 pixels. If the layout defines a text zone in a smaller size (e.g., header text), those texts are not displayed when Khmer is selected as the mobile phone language. In some cases, there can be a separate layout for the Khmer language.

3.1.1 Idle layout

The idle state is the basic state of the device. The device goes to idle after startup.

	Layout	Main zone	Softkeys*
128 x 160		128 x 115	LSK: 34 x 16 MSK: 52 x 16 RSK: 34 x 16

* LSK = left softkey, MSK = middle softkey, RSK = right softkey

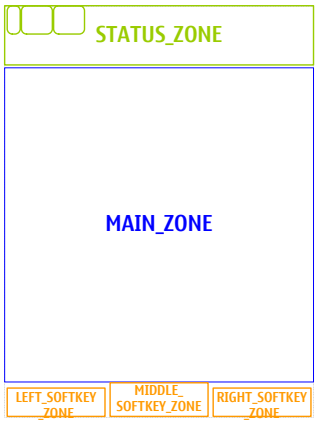
	Layout	Main zone	Softkeys*
240 x 320		240 x 246	LSK: 76 x 22 MSK: 76 x 26 RSK: 76 x 22

Table 2: Idle state layouts

3.1.2 Base layout with a scroll bar

Table 3 shows a base layout to display scrollable elements, for example, lists, grids, and data viewers.

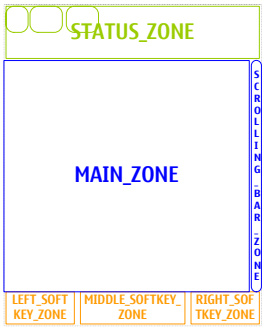
	Layout	Main zone	Softkeys
128 x 160		122 x 115 Scroll bar zone on the right side of the main zone.	LSK: 34 x 16 MSK: 52 x 16 RSK: 34 x 16
240 x 320		224 x 244 Scroll bar zone on the right side of the main zone.	LSK: 76 x 22 MSK: 76 x 26 RSK: 76 x22

Table 3: Base layouts with a scroll bar

3.1.3 Base layout without a scroll bar

Table 4 shows a base layout to display, for example, multimedia playing, image viewing, and text input.

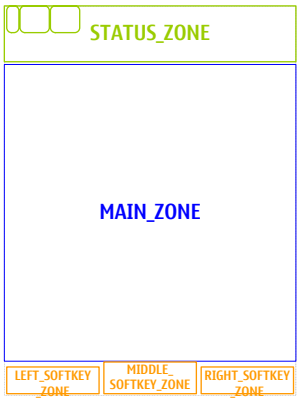
	Layout	Main zone	Softkeys
128 x 160		128 x 115	LSK: 34 x 16 MSK: 52 x 16 RSK: 34 x 16
240 x 320		240 x 244	LSK: 76 x 22 MSK: 76 x 26 RSK: 76 x 22

Table 4: Base layouts without a scroll bar

3.1.4 Browser layout

Detailed guidelines on how to optimize browser content to be shown in the Series 40 UI are available from the Forum Nokia Browsing technology pages: http://www.forum.nokia.com/browsing.

	Layout	Browser main zone	Softkeys
128 x 160		126 x 128 The user can choose to display a Web page title, which changes the main zone size to 126 x 115. The browser determines when the vertical and horizontal scroll bars are displayed, depending on the content and configuration.	LSK: 34 x 16 MSK: 52 x 16 RSK: 34 x 16

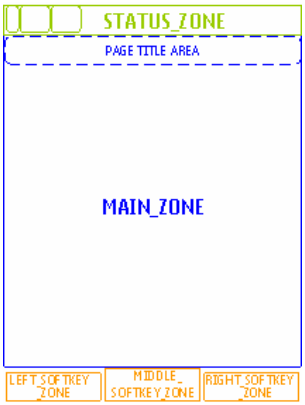
	Layout	Browser main zone	Softkeys
240 x 320		240 x 270 The user can choose to display a Web page title, which changes the main zone size to 240 x 246. The browser determines when the vertical and horizontal scroll bars are displayed, depending on the content and configuration.	LSK: 76 x 22 MSK: 76 x 26 RSK: 76 x 22

Table 5: Browser layouts

3.1.5 Java application layouts

Detailed guidelines on how to optimize Java application content to be shown in the Series 40 UI are available from the Forum Nokia Java technology pages: http://www.forum.nokia.com/java.

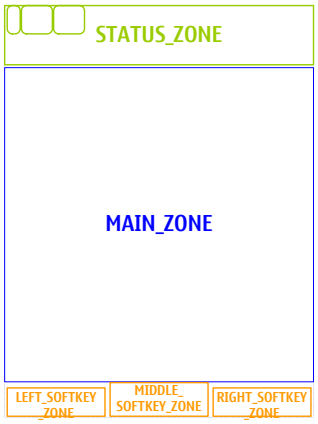
Design guideline: It is not recommended to optimize a MIDlet canvas to one display resolution only. If, however, this is done, and the MIDlet is being run in a product that has a different display resolution from the canvas defined in the MIDlet, the MIDlet is shown on the display according to the following generic rules:

When the device's display resolution is smaller than the canvas defined in the MIDlet, only the upper-left part of the canvas, which fits the display, is shown.

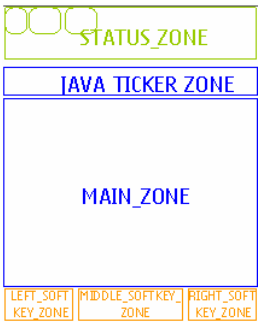
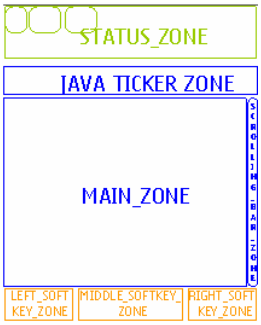
When the device's display resolution is larger than the canvas defined in the MIDlet, the canvas is shown in the upper-left corner of the display.

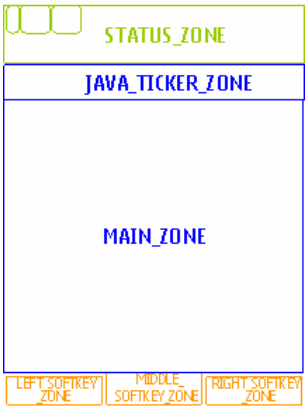
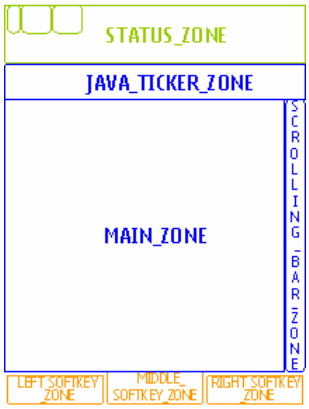
3.1.5.1 Java canvas normal

	Layout	Java main zone	Softkeys
128 x 160		128 x 115	LSK: 34 x 16 MSK: 52 x 16 RSK: 34 x 16

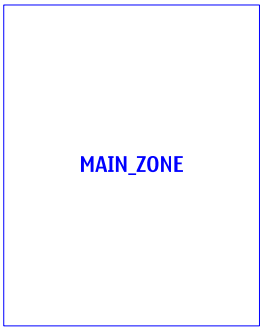
	Layout	Java main zone	Softkeys
240 x 320		240 x 248	LSK: 76 x 22 MSK: 76 x 26 RSK: 76 x 22

3.1.5.2 Java canvas with ticker

	Layout	Java main zone	Softkeys
128 x 160	Normal canvas: 	Normal canvas layout: 128 x 95	LSK: 34 x 16
	Form with ticker: 	Form ticker layout: 121 x 95	MSK: 52 x 16 RSK: 34 x 16

<table>
<tr><td rowspan="2">240 x 320</td><td>Normal canvas:

</td><td>Normal canvas layout:
240 x 218</td><td>LSK: 76 x 22</td></tr>
<tr><td>Form with ticker:

</td><td>Form with ticker layout:
223 x 218</td><td>MSK: 76 x 26
RSK: 76 x 22</td></tr>
</table>

3.1.5.3 *Java canvas full*

	Layout	Java main zone	Softkeys
128 x 160		128 x 160	- - -
240 x 320		240 x 320	- - -

3.2 Layout design guidelines

When creating layouts, the following issues should be taken into consideration:

- **User experience:** Design should create a positive user experience through the intuitive placement of items.
- **Simplicity:** There should be a limited number of font and graphic variations in a single layout and between different views.
- **Consistency:** The layout should be consistent with the general Nokia UI look and feel, both within the entire UI and between different UI styles.

Samples of good and poor layouts are presented below.

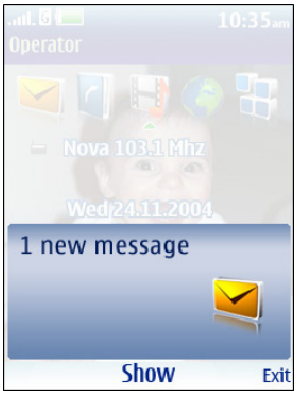
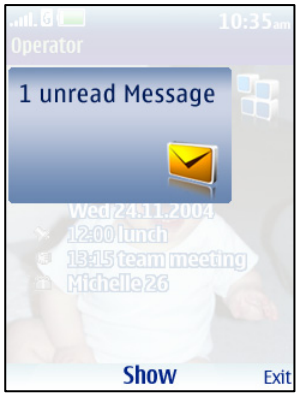
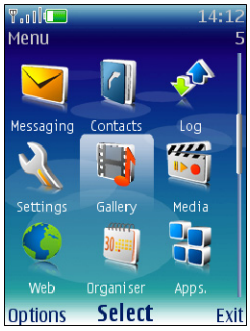
Good	Poor
 The layout is in balance, and the width of the screen is fully used for the notification text.	 The notification text area is not aligned in a balanced way with the layout, and the full width of the screen is not in use.
 Navigating between virtual buttons is easy because the buttons are placed in a row, close to each other. The eye moves naturally from one button to the next.	 Navigating between virtual buttons is difficult because they are not aligned in a row and are not close to each other. Extra effort is required to locate the desired button.

Table 6: Examples of good and bad layout design

3.2.1 Icon sizes

The icon sizes used in the layout design should be consistent. The main icon sizes used in different displays are listed in Table 7.

Icon type	Icon size	Notes	Example
Grid menu	128 x 160: 40 x 32 240 x 320: 74 x 62	Used in grid main menu.	

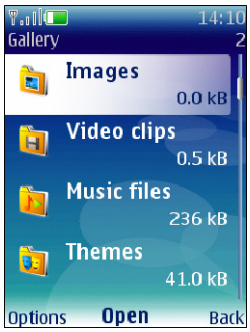
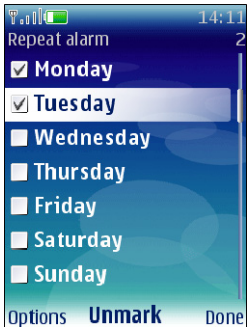
Icon type	Icon size	Notes	Example
List large	128 x 160: 21 x 28 240 x 320: 43 x 43	Used, for example, in second-level menus and in the tab menu.	
List	128 x 160: 16 x 16 240 x 320: 24 x 24	Used, for example, in third- or lower-level menus and data lists.	

Table 7: Icon sizes

3.3 Fonts

Most of the Series 40 fonts are proportional. A proportional typeface uses varying widths for characters, thus having proportional font means that it is difficult to tell how many characters will fit on each line. For example, the character “w” is much wider than the character “i.”

The font sizes (in pixels) and styles used in basic UI components are listed in Table 8. The font used in the 128 x 160 display is a bitmap font, whereas the 240 x 320 display uses a vector font.

Font	128 x 160	240 x 320
Softkeys (3 SK)	12 plain (LSK & RSK) 16 (MSK)	20 (LSK & RSK) 24 (MSK)
Header text	12 condensed plain	20
Main menu list	16	24
Submenu items	16	24
Option lists	16	24
Editors	16	24
Form labels	12 plain	20
Form field values	16	24
Messaging small	9	20
Messaging medium	16	24
Messaging large	23	28

Font	128 x 160	240 x 320
Browser x-small	-	12
Browser x-small bold	-	12 bold
Browser x-small italic	-	12 italic
Browser small	9 plain	16
Browser small bold	9	16 bold
Browser small italic	9 italic	16 italic
Browser medium	12 plain	20
Browser medium bold	12	20 bold
Browser medium italic	12 italic	20 italic
Browser large	16	24
Browser large bold	16 bold	24 bold
Browser large italic	16 italic	24 italic
Browser x-large	16	28
Browser x-large bold	16 bold	28 bold
Browser x-large italic	16 italic	28 italic
Browser xx-large	-	32
Browser xx-large bold	-	32 bold
Browser xx-large italic	-	32 italic
Java small	9 plain	16
Java small bold	9	16 bold
Java small italic	9 italic	16 italic
Java small bold italic	-	16 bold italic
Java medium	12 plain	20
Java medium bold	12	20 bold
Java medium italic	12 italic	20 italic
Java medium bold italic	-	20 bold italic
Java large	16	24
Java large bold	16 bold	24 bold
Java large italic	16 italic	24 italic
Java large bold italic	-	24 bold italic
Java form labels	12 bold	20
Java form field values	12	24

Table 8: Font sizes (in pixels) and styles used in basic UI components

Possible x-small, xx-small, x-large, xx-large, and so on, font sizes are mapped to the listed fonts above.

There are some exceptions to Table 8 in 128-width resolution because some languages do not support all font sizes.

- The Chinese font has two heights: 12 and 16 pixels.
- The Khmer font has two heights: 16 and 23 pixels.
- The minimum font height for some other non-Latin languages (for example, Telugu, Thai, and Sinhalese) is 12 pixels, whereas the minimum font height for the Latin font is 9 pixels.

When different scripts have been defined to be shown in the UI, and the language does not support the currently requested font size, the closest available font size is always used. For example, the following special rules are used:

- When the application requests a 9-pixel font size, the 12-pixel font size is used for Chinese and some other non-Latin languages. The Khmer language uses 16 pixel fonts in this case.
- When the application requests a 12-pixel font size, the 16-pixel font is used for the Khmer language.
- When the application requests a 23-pixel font size, the 16-pixel font is used for the Chinese language.
- If the layout for the text is such that the font size does not fit, then the text is not displayed at all (for example, if the header text layout is drawn for a 12-pixel font, the16-pixel Khmer text is not displayed as header text).

Resolution 240 x 320 has no language-specific exceptions.

3.4 Presentation of text

The following guidelines are used to display text and numbers on the display:

- In left-to-right languages, the default alignment is top left. In right-to-left language layouts, the default alignment is top right.
- Exception to the default alignment: In data queries with number mode only and in the calculator, the number alignment is from bottom right (in left-to-right languages).
- Word wrapping is recommended. Word wrapping means that if the character string does not fit the line, it is moved to the next line. If the string is too long to fit in an entire line, it is cut and displayed in several lines.
- Numbers should be truncated from the beginning and text from the end. Even if word wrapping is in use, truncation should be done so that as many characters as possible are displayed on the screen. Truncation is indicated by adding an ellipsis (...).
- Localized text should be displayed in sentence case by default, that is, the first character of the first word is uppercase (unless the conventions of the language require something different).
- Text and numbers should be displayed as entered by the user, without modifications (exception: number grouping for phone numbers).
- Nonsupported characters are represented with a box symbol. It is important that the original data remains unchanged and unsupported characters are not removed from text; each unsupported character is represented with a box.

3.4.1 Marquee scrolling

If text is truncated, it can be automatically horizontally scrolled to the opposite direction of the reading. There is a timeout before marquee scrolling starts, and the scrolling can be repeated. When the initial position of the text is displayed (after scrolling it once), the scrolling stops, and after a

timeout, scrolling starts again. If the text that is scrolled is focused (for example, in a list) and the user scrolls away from the item, the initial state of the item is restored and scrolling is stopped.

Layout limitations are one reason for using marquee scrolling.

Design guidelines:

Marquee scrolling should be used for user data, not for localized texts.

In lists, it is recommended that only one text is scrolled at a time and that the scrolling is only done when the item is highlighted.

Scrolled data should not be too long. With very long text, the user may forget the beginning of the text by the time the end of the text is reached.

Smooth scrolling is recommended, meaning that the text is scrolled continuously without perceivable stops during the scrolling cycle.

3.5 Graphical elements

The main graphical elements are:

- Icons
- Highlights
- Transitions
- Animations
- Fonts
- Scroll bars
- Pop-ups
- Themes
- Colors

When creating graphical elements, the objectives are:

Enhancing the user experience

- All graphical elements should contribute to usability.
- Graphical elements should not compromise the user experience by slowing down usage or mobile phone performance.
- Colors can be used to emphasize information.

Simplicity

- Graphical elements are not used as mere decoration (exception: themes).
- The user interface should not be compromised by, for example, icons when textual information is more relevant.
- Graphical elements should not interfere with primary information delivery on the display.
- Colors should be used in a harmonious way; excessive use of color should be avoided.

Creating a WOW effect

- In addition to enhancing the user experience, graphical elements like transitions can be used to create a WOW experience.

Consistency

- All graphical elements should be aligned with the general Nokia UI look and feel.
- Use of graphics should be consistent throughout the entire UI.
- Graphical elements should be coherent between different UI styles.

Cultural relevance and respect

- Use of colors and symbols needs to be carefully considered and reviewed.
- Cultural checks for possible inappropriate connotations should be performed.

4 Keys

The keypad is a set of physical keys. Series 40 technology achieves a small key count by allowing some keys to have more than one function, which varies according to the device's state. These keys are referred to as softkeys. Softkey text labels can be easily localized into numerous languages and they allow for a flexible UI style, which can be used in the same way in a variety of contexts.

The physical keys in Series 40 devices are as follows:

- Left softkey (LSK)
- Middle softkey (MSK)
- Right softkey (RSK)
- Up, down, left, and right navigation keys (scroll keys)
- Send key
- End key
- Numeric keys (0 - 9, *, #)
- Power key (optional)
- Volume up and down (optional)

As much as possible, the user should be able to use the device with just the navigation keys and softkeys.



Figure 3: Series 40 keypad

Softkeys are labeled with textual softkey labels on the display. The labels are shown close to the physical keys.

The navigation keys can be ordinary buttons (scroll keys) or they can be implemented by different mechanical solutions. Hardware solutions may have some effects on the navigation functionality. For example, a long key-press event may not be possible in all mechanical solutions.

The above listed keys are essential for the UI style. Additional keys may be added for certain application needs. For example, the following keys may be used together with Series 40 UI styles:

- Camera key
- Voice key for voice feature control (voice recorder, voice dialing, voice commands)
- Push-to-talk key
- Media keys (such as play, forward, reverse)

4.1 Key press

A key press is a press (down) and a release (up) of a key.

- In a short key press the key is pressed down for less than or equal to 0.8 seconds.
- In a long key press the key is pressed down for longer than 0.8 seconds.

The short key-press action is performed at the moment the key is pressed down. The long key-press action is performed at the moment the key has been pressed down for 0.8 seconds. If the pressed key

has only a short key-press action, only that action is performed no matter how long the key is pressed down. Depending on the key, the application, and the context, the long key press may result in a key repeat function. This means that the long key-press action will be performed as long as the key is held down starting after the key is pressed down for 0.8 seconds. Key repeat is used, for example, in the following situations:

- Character clearing
- List scrolling
- Grid scrolling
- Cursor movement in editors

An inactive key means that no action is assigned to the key press.

A key-press tone is generated when a key is pressed down. This happens even if the key is currently inactive and does not perform any action.

A long key-press tone is generated when a key has been pressed down for 0.8 seconds. If the key has no long key-press action, no long key-press tone is generated.

4.2 Softkeys

The MSK displays the primary function of the state. If applicable, the LSK can give more options. The RSK has the negative action (Exit, Back, Clear).

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

The default labels for softkeys are listed in Table 9.

State	Three-softkey UI
Idle state	Go to – Menu – Names
Selection lists	[Options] – Select – *Exit/**Back LSK Options is optional.
Multiselection lists	Options – Mark/Unmark – Back/Done RSK is Back when no changes are done. RSK is Done when changes are done. LSK Options includes items such as Mark all/Unmark all.
Confirmation queries	(empty) – Yes – No for two-option query No – Yes – Back for three-option query

State	Three-softkey UI
Data queries and editors	Options – OK/Search – Back/Clear MSK is OK by default. MSK is Search in empty data queries if the fetch service (from contacts) is available. LSK is Options when there are predictions or other editor functions available in the query editor. The Option list also possibly includes Search. RSK is Back when there are no characters entered to the data query. RSK is Clear when there are one or more characters entered to the data query.
Forms	Options – (field-specific MSK) – Back/Close MSK can be, for example, Change when the focus is on a field where the MSK is used for changing the value of the field. RSK is Back when there are no changes done in the form. Close RSK is used after changes have been made.
Viewers	Options – (Application-specific MSK) – *Exit/**Back
Interruption	[Options] – (Application-specific MSK) – Quit RSK is Quit when pressing it interrupts an ongoing operation. It is used with operations when it is not possible to guarantee that the original state can be restored. LSK Options is optional.
Stopping	[Options] – (Application-specific MSK) – Stop RSK can be Stop instead of Quit, if a familiar application using the Stop label is integrated in the device. LSK Options is optional.
Cancellation	[Options] – (Application-specific MSK) – Cancel RSK is Cancel when pressing it truly cancels an ongoing operation. It is used in situations where no permanent changes have been made yet and the original conditions can be restored. LSK Options is optional.
Soft notifications	[Options] – (Application-specific MSK) – *Exit LSK Options is optional.

Table 9: Default labels for softkeys

Design guideline:
* The Exit label is used as the RSK text when pressing it brings the device to the idle state or when the application is going to be closed in Java MIDP.
** The Back label is used as the RSK text when pressing it returns the device to a previous state.

4.3 Scroll keys

The up and down scroll keys are used for scrolling through lists, such as the menu, selection lists, and option lists, to reach the functions and data of the device. In character entry states, all scroll keys are used to move the cursor. In data viewing states, scroll keys are used to move to the next or previous view. By default, the scroll keys implement key repeat, meaning that the scrolling action is performed consecutively while the scroll key is kept pressed down. See also Section 5.3, “Navigation.”

If the device does not have volume keys, the volume is controlled with the scroll keys.

4.4 Send key

The Send key is used for answering incoming calls and creating outgoing calls. For other features than call handling, the Send key can only offer a shortcut, which means that the function assigned to the Send key needs to be offered in the standard way, too. The shortcut function assigned to the Send key need to be somehow network operation related, for example, sending data to the network or fetching data from the network.

Note: The Send key is not used as a general selection key. Exceptions are Java MIDlets and the browser, where the select function can be duplicated to the Send key.

4.5 End key

The End key is used for terminating ongoing calls and network connections. It is also used for rejecting incoming calls.

If there is more than one call, the End key press terminates the active call. When there are no calls or network connections, the End key press terminates and exits the current application and returns to idle.

Note: In the mobile information device profile (MIDP), the End key closes the application when there are no calls or network connections.

When the user is using applications during a call, the End key press terminates the call and doesn't exit the application. When the user is using an application with network connection, the End key press terminates the connection and doesn't exit the application if it is feasible to continue the application without the connection.

Note: A long key press of the End key always exits to idle.

If the device does not have a separate power key, the End key is used for powering the device on/off. Power control in the End key is done in a way that all the above-described functionality is not changed.

4.6 Numeric keys

The main use of the numeric keys is for numeric and alphanumeric character entry.

The very basic principles of numeric key functionality are the following:

- In idle: enter digit and start number entry application.
- In editors: enter character.
- In menus, lists, viewers: application-specific function or inactive. For example, activate search function in Contacts list and menu shortcuts.
- During a call (in the call status screen): enter digit and start number entry application.

Note: A long press of the 0 key in the idle state launches the browser.

4.7 Power key

The device can be turned on and off by a long key press of the power key.

When the device is on and in idle, a short key press of the power key displays the power key menu, which has options to switch off the device and set the current profile.

4.8 Volume keys

Volume keys are used for adjusting the speaker volume and audio volume in general. Volume key up is pressed to increase the volume level; volume key down is pressed to decrease the volume level.

During a voice call, the volume keys always adjust the voice call volume. In audio applications, the volume keys adjust the audio volume. When in idle, the volume keys adjust the voice call volume, unless there is an audio application in the background, in which case the volume keys adjust the audio volume.

4.9 Keyguard

Keyguard is a feature that allows users to lock the keys. This means that the device does not react to any key presses, except for the key-press sequence used for unlocking the keys. Other key presses just generate an information note instructing the user to unlock the keys.

Note: The keyguard is activated and deactivated by the softkey Menu/Unlock + star key (*) key-press sequence entered within a timeout.

When the keyguard is active, the keyguard status indicator is shown.

The keyguard is deactivated when there is an incoming call and reactivated after call termination.

When the keyguard is active, the device still needs to recognize the emergency number input key-press sequence and enable emergency calls.

5 Interaction

5.1 Idle state



Figure 4: Idle state

The idle state is the basic state of the device. The device goes to idle after startup.

In the idle state the device is waiting for input from a user or network. The idle state design is driven by a sense of aesthetics and tranquility. When using the device, the user is able to personalize the idle state appearance, but the default appearance of the idle state is a classic, harmonious Nokia style.

In the idle state, the display shows the following:

- Signal bar
- Battery bar
- Soft indicators
- Soft notifications
- Status indicators
- Clock

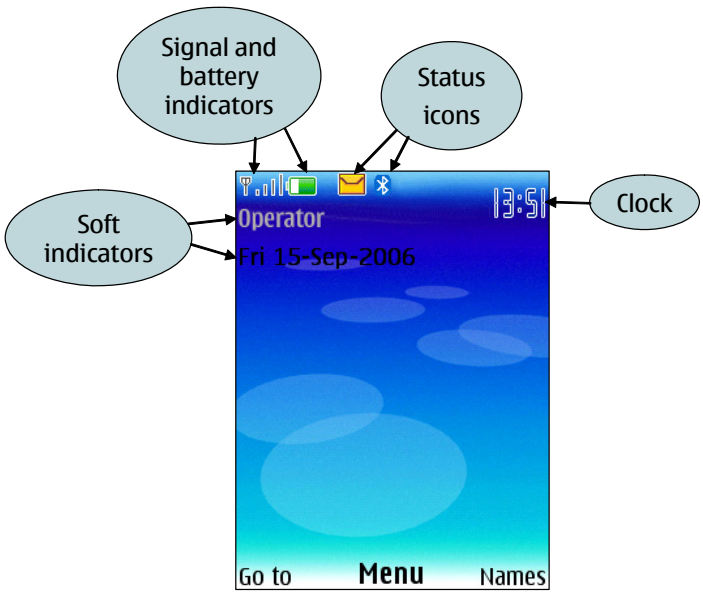


Figure 5: Elements shown in the idle state

Soft indicators are textual indicators displayed in the idle state that inform the user of the device status, for example by showing the operator name or what profile is active. Soft indicators may contain a graphical icon. If more than one soft indicator is shown at a time, they are displayed according to their relative priorities.

Note: The LSK is Go to (by default), the MSK is Menu, and the RSK is Names (by default) in the idle state.

5.1.1 Idle personalization

The user can personalize the background image (wallpaper) and font color of the idle state. In addition, the user can also personalize key-press shortcuts for the idle state. For example, the left and right softkeys and scroll keys can be personalized.

The active standby feature can be used to personalize the main zone contents for the idle display. With active standby it is possible to add, for example, a shortcut bar (containing links to favorite applications), music player, today's calendar notes, or the user's own textual note on the idle display.

In the Series 40 UI style, none of the active standby parts are in focus in the idle state. The user can enter numbers and instantly create a call from idle or activate the keyguard. The MSK is used for main menu access and thus cannot be used to select an object on display. Pressing a scroll key from the idle state does the activation of the active standby parts. Activation changes the UI state, and the device is no longer in the idle state. The user can change the activation keys of active standby.

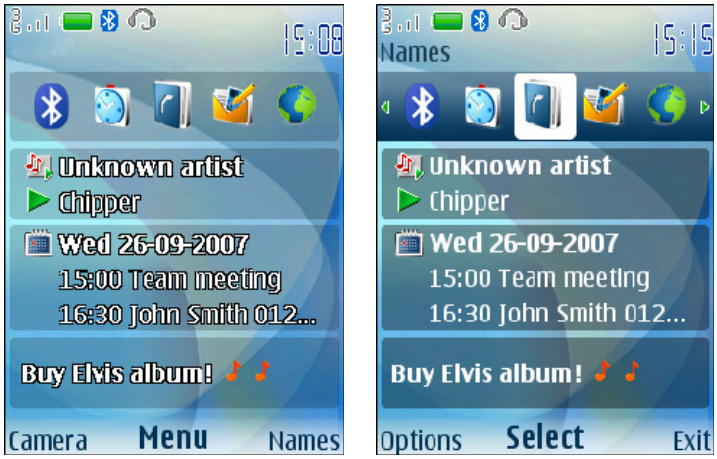


Figure 6: Series 40 active standby – left display is inactive mode, right display shows active mode

5.2 Menu

The Series 40 UI style is based on a hierarchical menu structure. The applications, functions, and data of the device can be reached via a menu interface. The menu contains a main menu and submenus. The structure of the main menu and each submenu is a closed loop, meaning that the first item is reachable from the last item by scrolling forward, and the last item is reachable from the first item by scrolling backward.

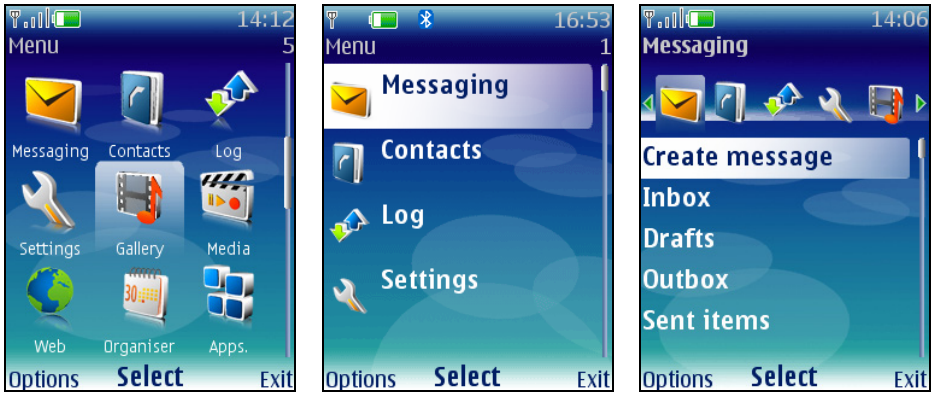


Figure 7: Series 40 UI menu style examples – grid, list, and tab menu (tab only in 240 x 320 resolution displays)

5.3 Navigation

Figure 8 shows an example of a basic navigation hierarchy. The solid blue lines indicate moving forward from a state into a substate. The red dotted lines illustrate backstepping to the previous level. Items inside yellow ellipses can be scrolled in a closed loop.

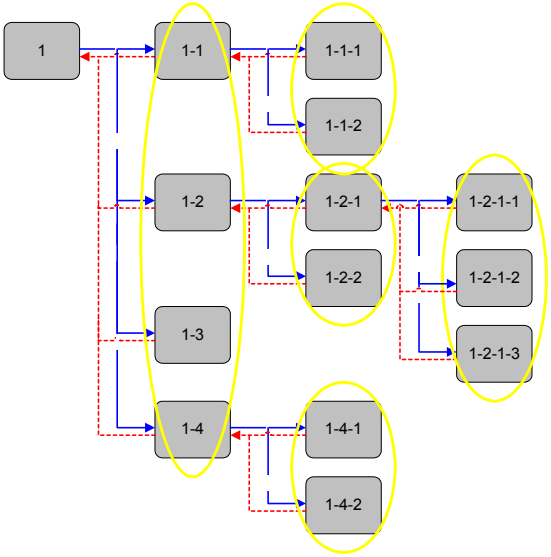


Figure 8: Example of a basic navigation hierarchy

Design guideline: As much as possible, users should be able to operate the device using only the scroll keys and softkeys. The MSK is used for selection and going deeper in the menu; the RSK is used for backstepping.

5.3.1 Four-way navigation

Navigation is done with scroll up, scroll down, scroll left, and scroll right keys. The default functions of the scroll keys are the following:

Scroll up:

- Moves the focus one step up in lists and grids.
- Moves the cursor one line up in editors.
- Scrolls one page up in data viewers.

- Moves the focus to the previous element above and/or scrolls the window upwards in the browser and MIDP.
- Increases a data value by one step (for example, in time entry).

Scroll down:

- Moves the focus one step down in lists and grids.
- Moves the cursor one line down in editors.
- Scrolls one page down in data viewers.
- Moves the focus to the next element below and/or scrolls the window downwards in the browser and MIDP.
- Decreases a data value by one step (for example, in time entry).

Scroll left:

- Moves the focus one step left in grids.
- Moves the cursor one character left in editors.
- When there are no items on the left, moves to the right edge of the previous/next line (depending on the language reading order).
- Moves the focus to the element in the left and/or scrolls the window left in the browser and MIDP.
- Decreases the speaker volume by one step during a call (in case there are no separate volume keys).

Scroll right:

- Moves the focus one step right in grids.
- Moves the cursor one character right in editors.
- When there are no items on the right, moves to the left edge of the next/previous line (depending on the language reading order).
- Moves the focus to the element in the right and/or scrolls the window right in the browser and MIDP.
- Increases the speaker volume by one step during a call (in case there are no separate volume keys).

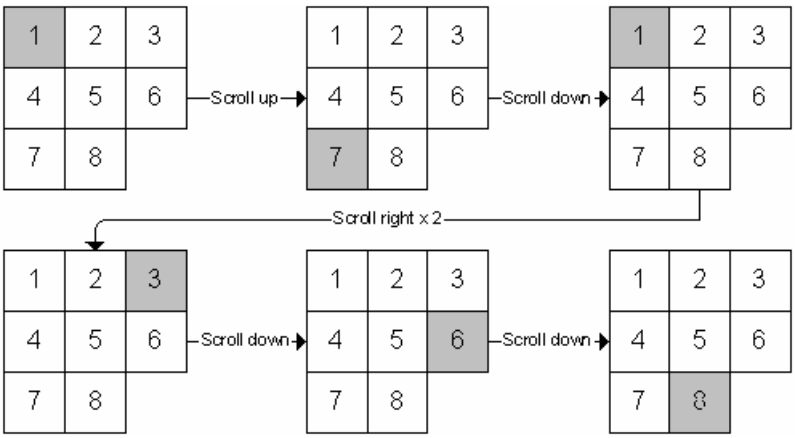


Figure 9: Example of scrolling a grid with a four-way scroll

5.4 Navigation aids

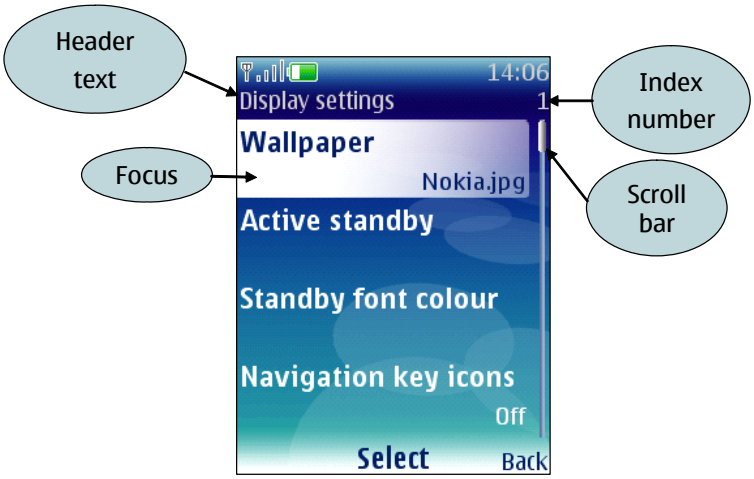


Figure 10: Navigation aids in the Series 40 UI

5.4.1 Focus

The current position in a list or grid is indicated by focusing the current item. Focus is indicated by highlighting the item (highlight bar). Items occupying the full screen are not highlighted, but the item shown on the screen is the currently focused one.

The subsequent operations (for example, selection, open options) concern the focused item.

5.4.2 Header text

The header is a text aid that helps the user identify the current location in the application or menu structure. The header text is positioned at the top of the display (status zone).

Typically the header text presents the title of the previous menu or application level, or it presents the title of the data that is being processed.

The header is used in lists, grids, and viewing states.

5.4.3 Scroll bar

The scroll bar assists the user in navigation by expressing what portion of the data is visible on the display. The scroll bar is a vertical bar with an indicator element, a thumb, and it is displayed at the right edge of the display. The thumb moves up and down along the bar as the list position changes.

There may also be a vertical scroll bar visible at the bottom of the screen, above the softkeys, in two-dimensional views.

With the help of a scroll bar and the thumb, the user can detect the first and last items of the list and recognize the current position by comparing the thumb distance between the top and bottom of the scroll bar.

The scroll bar is used in lists, grids, and viewing states.

5.4.4 Index numbers

In addition to the scroll bar, index numbers are used to indicate the position of the currently focused choice item in lists and grids.

There are two types of index numbers: menu index numbers and dynamic index numbers.

Menu index numbers are used only in the menu — the numbers represent menu shortcuts from the idle state.

- If items are temporarily hidden, menu index numbers are not recalculated.
- Menu index numbers are not used for option lists.

Dynamic index numbers are used in option lists (except in pop-up option lists), data lists, and multiselection lists.

- Dynamic index numbers are recalculated each time a list is opened.
- They are not shown if a list is empty.

5.5 Backstepping

Design guideline: The back-stepping possibility should always be offered to the end user to minimize the effects of an error in the key-press selection.

Backstepping is a different concept from “return after successful or unsuccessful operation.” Backstepping is always performed by the user, and it means that the operation was not completed by the user.

The basic backstepping rule is that backstepping returns to the previous stage in the application and operation hierarchy. The rule also applies to option lists.

After backstepping to the previous state, the screen looks the same as before. The focus is returned to the same items, and items are kept in the same positions on the screen.

When an application activates another application, the mobile device backsteps to the starting application when returning from the other application.

Design guideline: If a backstep function on a softkey actually performs an exit to the idle state, or a Java application is going to be closed, the softkey is labeled as Exit instead of Back.

The following states in the UI are not backstepped to:

- Soft notifications
- Confirmation queries
- Security code queries
- Notes

Note: The exit function to the idle state is always available in the long press of the End key.

5.6 Text input

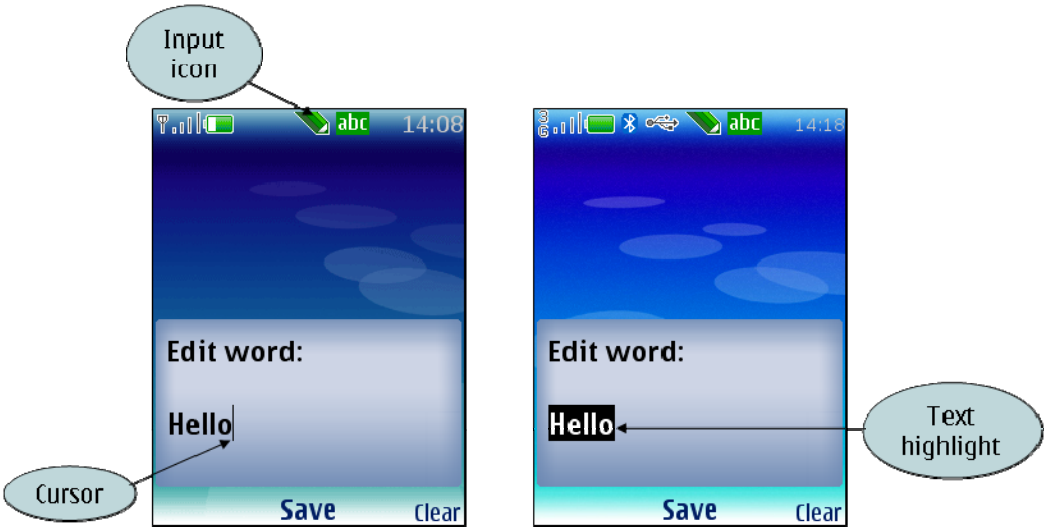


Figure 11: Examples of text input screens

Forms, editors, and data queries enable user data entry. Text is entered by pressing the numeric keys. When the editor is open, a general input status indicator is shown. The interaction of the editor is consistent in all instances, but the visual appearance may be one of the following: form text editor, full-screen editor, data query editor, or number entry editor.

In most cases it is possible to enter more characters in the editor than can be shown at once on the display. In such cases, only the range of data that fits the display is shown, and the rest is hidden.

If the length of the data is restricted, and it can be assumed that the user quite often encounters this limitation, it is recommended that the user be informed about the maximum length of the text. In any case, where the data length is restricted, the user cannot enter more characters than the maximum length. After the maximum length has been reached, the character keys are inactive.

A blinking vertical cursor is shown at the insertion point. The user can move the cursor with the scroll keys, as described in Section, 5.3, “Navigation.”

There are two ways to enter characters: the traditional multi-tap method and prediction input. The multi-tap method requires many key presses per character due to the multiple mappings of characters to each key. This method is the initial default for all editors.

Predictive text input (character or word prediction) provides a more efficient way to enter text than the traditional multi-tap method. In predictive input only one key press per character is needed. This is possible because of an intelligent dictionary that allows the editor to predict what word the user intends to enter based on the number and combination of key presses. There are separate dictionaries for different languages.

The editor can be in either an alpha mode or a number mode. In the alpha mode there may be different character cases and input methods available depending on the current input language. The numbers of the keys can be entered with a long key press. Also in the alpha mode, there are special characters under the "1" key and in the special character screen. The special character screen is accessed by pressing the * key. The user can select one of the characters to be inserted to the text.

In the number mode, only digits and certain special characters are available. There exist different number modes, which the applications can use based on the needs. For example, plain number mode contains only digits, but for entering phone numbers, the number mode also contains +, #, *, p, and w characters.

The user can delete characters one by one with RSK Clear. By doing a short key press of Clear, one character is deleted, and by making a long key press, several characters are deleted at an accelerating speed.

Text can be highlighted, either by the user (when selecting text for copying and pasting) or by the phone automatically when text is being entered in the editing screen. When text is highlighted by the phone, entering any character removes the highlighted text and replaces the entry with the inserted character. Pressing the Clear softkey removes the highlighted text with one key press. The highlight can be removed without replacing the text by pressing the scroll keys.

Design guideline: Automatic highlighting of text should be used when there is a need to display a predefined text but there is a certain likelihood that the user will clear the text from the input and enter new text. Automatic text highlighting can be used, for example, when the user renames a file name that is generated by the phone.

5.7 Feedback

Immediate feedback should be given to the end user once the user has selected an action. This provides users with a sense of control and a feeling of progress and accomplishment.

Feedback can be visual or a combination of visual and audio. Feedback reflects the current status. The text is context sensitive. Graphics are used to alert the user to what type of information is displayed; for example, an error is more alarming than a confirmation note. Audio feedback is used to get the user's attention.

Different types of user feedback are described in the UI components chapter, for example, in Section 8.9, "Confirmation and list queries," Section 8.10, "Soft notifications," and Section 8.11, "Notes."

6 Sounds

The use of tones strongly shapes the image of a product, and therefore audio design has some important general guidelines. Sounds should be understandable, simple, clear, and pleasing to the ear. The tone should be delightful, not annoying. The user should be able to interrupt any tone with a key press, and the length of a tone needs to be appropriate for its intended meaning.

The following four goals should guide audio (tone) design:

- Functionality: All sounds should have a function, and the sound should serve that function well. Each sound should improve the usefulness and usability of the product. Excess auditory feedback should be avoided in the standard UIs.
- Consistency: Tones used for similar purposes in separate contexts should resemble one another.
- Limited sound set: The power of sounds and learnability may be destroyed by excessive use of sound.
- Personalization: Users should be able to customize the sound palette and sound usage.

There are two basic categories for different tones:

- Notification tones are used to attract the user's attention, even when the user isn't concentrating on the device. Thus, notification tones need to be audible from a distance. Notification tones are either informative or demanding, depending on the nature of the situation. Urgency can be communicated via the demanding element in the sound — for example, high for incoming calls and low for incoming text messages.
- Feedback tones are used to give immediate audible feedback about the user's actions. By definition, feedback tones are played when the user is holding and concentrating on the device, and thus, should not be ear-piercing. Feedback tones are used for confirmation and error communication, for example.

7 Localization

When localizing a UI, the following elements (at minimum) should be localized:

- Character sets: It is required that the UI be able to present and handle all characters in the supported character sets, no matter what the current UI language is. The UI needs to be able to show and edit all characters, both in viewing and editing states.
- Alphabetical order: Alphabetical order is language dependent.
- Layouts: Layouts are combined with the display text language setting. Alignment and text flow direction should be selected according to the rules of the language.
- Graphics: In general, the graphic icons, animations, and colors are the same for all language UIs, unless something is completely forbidden by a culture. In such cases, a separate version should be made for that particular language.
- Key mappings: The characters assigned to each numeric key depend on the text entry context and the current language used in the UI. In addition to the default input language defined by the display language setting, the user can change the input language in some editors via a separate setting.
- Predictive text dictionaries: Each display language supporting the predictive text input method has a separate dictionary.
- Date and time formats: There are default date and time formats and separators for each display text language. The user has the option to change the default values via a separate setting in time management applications.
- Start day of the week: The default start day of the week is dependent on the display language setting. The user can change the default value in the calendar application.
- Units of measurement: Units of measurement used in applications, for example, units representing a position should be localized, too.
- Tones and sounds: Tones should be appropriate to their cultures. Sounds that are very culture-specific should be avoided in the default tone set. It is possible, for example, to create localized ring tones for a certain culture or market area.
- Display texts: Display texts give information and feedback on the UI's functionality and states. Display texts are translated to all languages supported by the UI, with the exception of special cases with some customized, branded, or co-branded feature names.

8 UI components

To maintain consistent functionality across applications and situations in the UI, the UI is designed by using UI components. UI components are the building blocks used to build an application or a feature's functionality and appearance.

A UI component is an independent piece of the UI that can be used in many contexts. It has a well-defined interface outside and can be redesigned inside in such a way that no effects to the utilizing applications are needed.

The main benefit of using UI components is consistency — that is, the same functionality has a similar behavior and appearance in all contexts. Using UI components also increases efficiency by reducing overlapping design work.

Pop-up appearance

A UI component can be displayed as a pop-up window. A pop-up window is displayed on top of the original display, so parts of the current application are visible behind the pop-up window. This allows users to see the context of the item that they are setting.

Pop-up windows do not cover the status bar or softkey area. When a pop-up window is displayed, the background application is dimmed to emphasize that the user interaction is with the pop-up window and not with the application behind. The softkey labels are not dimmed since they apply to the pop-up window — if the pop-up window has no related softkeys, no softkey labels are displayed.

Pop-up windows can be layered. If a pop-up window is displayed when another pop-up window is already on the screen (for example if a pop-up options list is displayed and selecting the option leads to a note or a confirmation query) then the first pop-up is displayed dimmed in the background.



Figure 12: Example of layered pop-ups

The following UI components support the pop-up appearance:

- Data queries for multiple fields, such as date and time
- Confirmation and list queries
- Soft notifications
- Notes (including help texts and volume controls)
- Options lists
- Call pop-ups (for example incoming call, call active, outgoing call)

8.1 List types

Selection lists consist of one-row choice items or multiple-row choice items. Multiselection lists can be applied when there is a need to make several selections from the same set of options (see Section 8.3, “Multiselection ”).

Design guideline: The height of the list items should be the same, and if the list displays icons with text, all list layouts should contain the icon zone (even if the icon is not displayed).

All lists have a scroll bar, an index number, and header text, by default.

8.1.1 One-row choice items

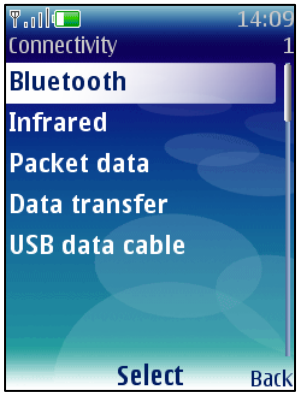


Figure 13: Example of one-row choice items

One-row choice items appear in menu lists, option lists, and data lists. The choice items mainly contain text, although they may also contain icons (for example, message lists). The primary icon of the choice item is located at the beginning of the text string. If a secondary icon is needed, it will be located at the end of the text string.

8.1.2 Multiple-row choice items

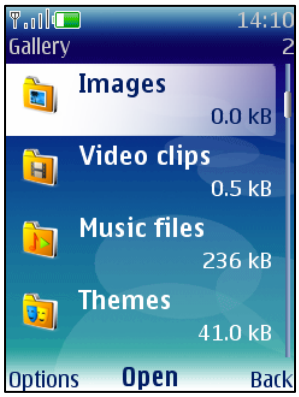


Figure 14: Example of multiple-row choice items

Multiple-row items appear in menu lists, data lists, and as setting choice items (see Section 8.2, “Settings”). Each item in the list always includes the same amount of rows. There can be exceptions in some data lists (for example, calendar notes list) where the item height can differ based on how much content the data item contains.

The choice items may also contain icons.

8.2 Settings



Figure 15: Example of a settings screen

Settings have a static part, which tells the name of the setting, and a dynamic part, which shows the current value of the setting. When a setting item is selected, a setting value list is opened with the current value focused.

The setting value list uses one-row choice items.

If the setting contains no value, the dynamic part displays “No value,” where the value equals the item name, for example, “No image.”

When the setting value is a user entry (browser settings, for example), selecting the settings item opens a data query.

8.3 Multiselection lists

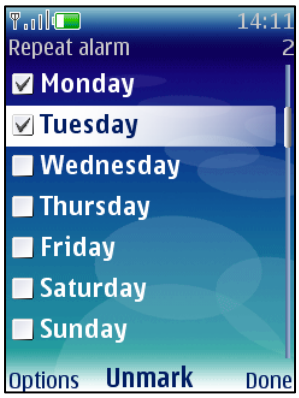


Figure 16: Example of a multiselection list

A multiselection list allows the user to select several items at once. Multiselection lists have the same appearance and behavior as one-row choice items.

The main difference is that a check box appears to the left of the multiselection item. When the item is highlighted, the user can “mark” or “unmark” the active item with a softkey. When the items are marked, the check box is checked. Any unselected items will have an unchecked box next to them.

The LSK gives users the options to mark and unmark all the items at once and view only selected or unselected items.

When the user has not done any changes in the multiselection list, the RSK is “Back,” and pressing it makes a return to the previous state.

After any changes, the RSK changes to “Done,” and pressing it displays a confirmation query allowing the user to save or ignore the changes.

8.4 Option lists

An option list displays options of a selection list item using one-row choice items.

The basic rule is to include all the available functions to the option list in the LSK. However, in order to keep the option list as short as possible, the functions of the other softkeys will not be repeated in the list. Also, functions that have dedicated hard keys are left out, as it is assumed that these are obvious to users.

For options, a maximum of two displays is recommended. Longer option lists should be shortened by grouping similar options together into a sublist, for example “Sending options.”

Design guidelines:

- The most-often-used options are put at the top of the option list and the less-frequently-used ones at the end.
- The order and grouping of options is consistent within the application and component.
- Similar options are put in sequence.
- Context-sensitive options are listed before application-wide or universal options.
- Application-wide and universal options should be avoided. Thus, the options should address the focused item. In particular, the deeper the user moves into the application structure, the fewer application-wide and universal options there are in the option list.

8.5 Grids

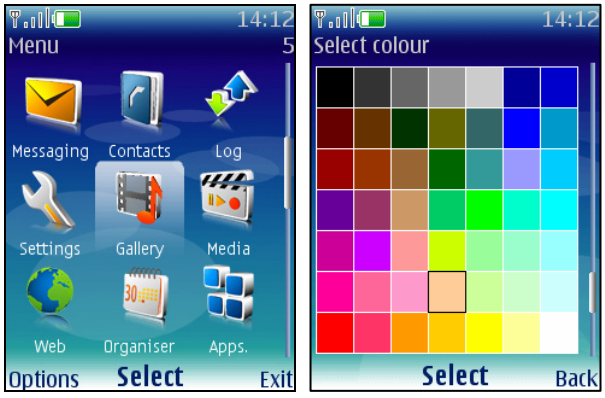


Figure 17: Examples of a grid

In a grid, the choice items are in a two-dimensional rectangular arrangement of cells. One cell is focused at a time.

Grids have a scroll bar that indicates the relative vertical location of the line where the currently focused cell is. Header texts and index numbers are used with grids. Index numbers run from the upper-left cell to the lower-right in left-to-right languages.

A grid is used, for example, in the calendar month view, in the special character screen, or as the alternative view in the main menu and in the Gallery application. (The calendar month view does not show the scroll bar or index numbers.)

The default grid filling order in left-to-right languages is:

1	2	3
4	5	6
7	8	9

The default grid filling order in right-to-left languages is:

3	2	1
6	5	4
9	8	7

For information on navigating in a grid, see Section 5.3, “Navigation.”

8.6 Tabs

Tabs can be used to present related data views in parallel. Tabs provide fast switching between the related views and are useful for categorizing and giving a quick overview of multiple data items. Tabs contain more than one tab page. Each page is indicated with a tab icon and a text label. The text label is displayed as header text. The tab page content (for example, a selection list) is displayed below the tab icons.

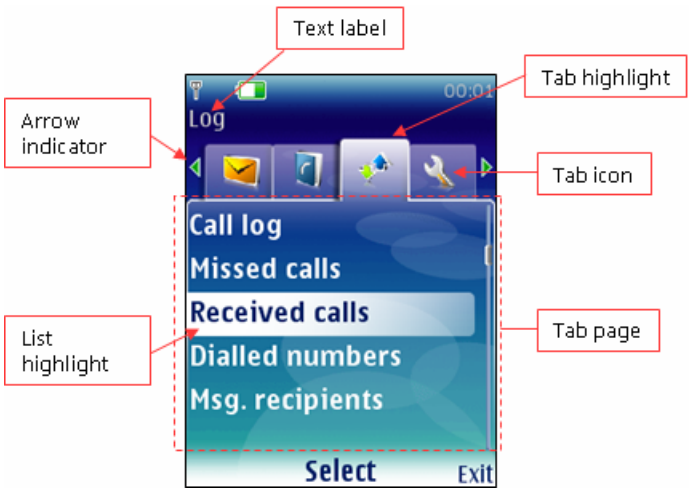


Figure 18: Example of tabs

When opening a view with tabs, the first (leftmost) tab is focused by default. Pressing the scroll left and right keys navigate between tab pages. The tab highlight moves to the left or the right when navigating the tabs. Tabs can be scrolled in a loop; thus scrolling to the right from the last tab moves the focus to the first tab and vice versa. Moving the tab highlight when looping from the last to the first tab, or from the rightmost position to the leftmost position, gives visual feedback to the user that the tabs are looped.

When there are more tabs than fits on the display, following rules are applied:

- Arrow indicators are displayed next to the last and first visible tab to indicate that there are more tabs.
- Tab highlight does not move to the rightmost or leftmost position until the last or first tab is reached. Thus, the tab highlight stays on the next to last position on the display when scrolling to the right and the focus is not moved to the last tab. The same rule applies when scrolling to the left.

  This navigation method gives the user visual feedback about when the last or first item is reached and gives the possibility to see the next tab in the scrolling direction.

Note: The scrolling rules and default highlight position are reversed for right-to-left languages.

Within the tab page, there can be another highlight (for example, list highlight) which is moved by scrolling up or down. When scrolling between tab pages, the tab page highlight position is reset, that is, always starting from the top of the list.

Tab page content does not need to fit on one display; it is possible to display a scroll bar on the right side to indicate the relative position within the tab page. The tabs are always kept on display, thus scrolling downwards from the last visible item on the tab page moves the tab content, keeping the top part (tabs and text label) in the same position on the screen.

The middle softkey (for example, Select) applies to the currently focused item on the tab page, not the entire tab. The left softkey can contain options that apply to the highlighted item, the tab page content, or the tab (for example, delete the tab, if that is possible in the application). It can also contain application-generic options (such as application settings). The right softkey exits the whole view with the tabs, and goes to the previous UI level before the tabs.

Design guidelines for using tabs in applications:

It is recommended not to have more tabs than possible to display on one screen.

Information shown in the tabs should be categorized in a way that matches the user's mental model of the data. The information that the user wants to see together should not be distributed across different tabs.

If the contents in the tabs are not equally important, the default tab should include the most important information.

Tab page content cannot be interacted with the scroll left and right keys. These keys should always navigate between tab pages.

Nested tabs (that is, tabs inside tabs) are not recommended, because the user can get easily lost in the tab's structure.

8.7 Data queries

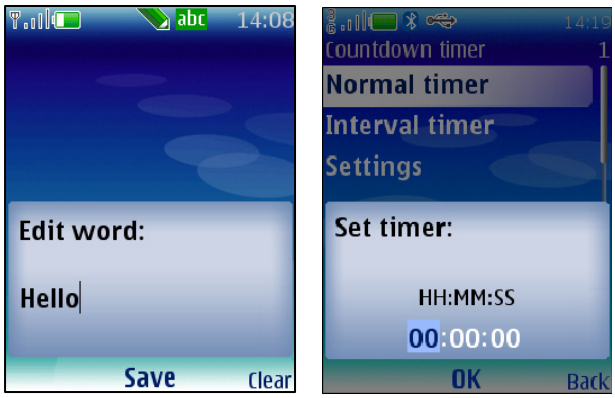


Figure 19: Examples of data query screens

Data queries are used to request user entries. Data queries have a prompt text and a user-entry space.

The OK label is used as the default, confirming text in data queries. The Search label is used in empty data queries instead of the OK text if the fetch service is available and it is not necessary to allow empty user entry.

The Options label is also used as the LSK text when there are prediction or other editor functions available in the query editor. The option list can include a search function if it is possible to search data into the query. If the user has entered data and uses the search function after that, the searched data overwrites the user entry.

The Back label is used as the RSK text when there are no characters entered to the data query. The Clear label is used as the RSK text when there are one or more characters entered to the data query.

In addition to the normal data query functions, date and time queries also enable four-way scroll keys: scrolling up and down can be used to decrease and increase the entry value, and scrolling right and left can be used to move to the next or previous entry field. The date and time queries also contain advice text that gives guidance about what time or date parts are queried (for example, “hh:mm:ss” means that the user enters hours, then minutes, and finally seconds).

8.8 Forms

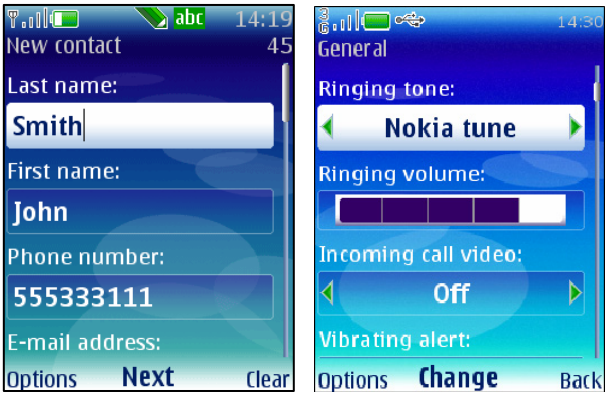


Figure 20: Examples of forms

A form is used to create and edit a set of data. It consists of a number of fields, each of which represents an item of data. A field consists of a field label, which is shown first to give textual information about the field, and a data part. The field label is optional, that is, a field can exist without a label.

For example, the following types of fields in forms can be supported:

- Text editor
- Date and time entry
- Selection or multiselection field
- Image field
- Multimedia field
- Level control field
- Static text field
- Virtual buttons

The user can use scroll keys to move within a form. Scrolling down and up moves focus to the next or previous field if there is nothing more to display in the current field. When the focus moves to the next or previous field, as much as possible of that field is displayed. Line-by-line scrolling is used for scrolling inside a text editor field that contains input that is longer than one line.

There is no looping in a form; for example, when the focus is in the last field it is not possible to move from the last field to the first field by scrolling down.

Left and right scrolling scrolls the cursor within a field, where applicable. The cursor is moved character by character. In cases where there are multiple fields in one row, the left and right scrolling moves between the fields. In selection fields and level control, scrolling left and right changes the value of the field. Scrolling left and right in a level control field increases/decreases the value by one increment. In selection fields, scrolling left/right displays the previous/next value in the field. The user can continue scrolling through the values in a loop; reaching the "last" value and scrolling to the right displays the "first" value (and vice versa).

The LSK in forms is Options if there are any additional options. The options list can contain options such as Save and Cancel, or field- and/or feature-specific options. The MSK includes the highest priority positive function (for example, changing the value or opening a sublist), and the RSK includes the highest priority negative function (for example, in a text editor field the Clear function has a higher priority than the Back function).

The MSK in forms contains a field-specific action. For example, for selection fields, the MSK is Change, which opens the value list. Selecting the value from the list returns the UI back to the form field.

There can be virtual buttons at the end of the form, after all the fields, which can be used for confirming and canceling the form. If possible, these buttons should be placed on the same line.

8.9 Confirmation and list queries

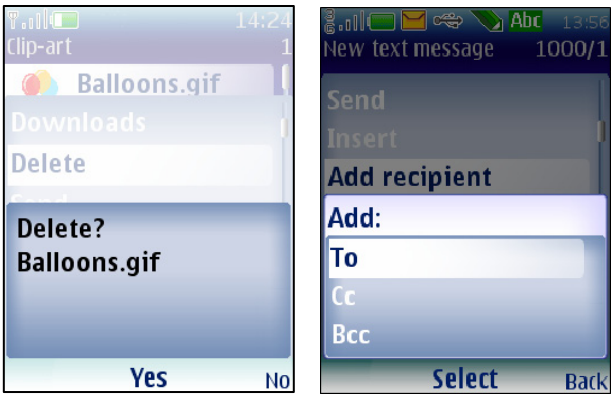


Figure 21: Examples of confirmation and list queries

A confirmation query or a list query is used when the UI requires the user to select between alternative choices in order to proceed with an operation. What happens after the query has been confirmed or discarded depends upon the application.

The confirmation query forces the user to choose between accepting and declining an operation. The choices are given in the softkey labels. In the confirmation query, declining and backstepping possibilities can be offered separately. In this particular case, the declining action is mapped to the LSK, which is not usually allowed, and the backstepping action is mapped to the RSK.

Confirmation queries display a question to the user that can contain the name of the object (for example, file name) that the query concerns. That is, confirmation query texts are context sensitive. If the entire text does not fit into the confirmation query layout, the confirmation query is scrollable.

Design guideline: Confirmation queries are recommended for operations that can be harmful to the user (for example, deleting data).

The list query is used when there are more alternative choices than can be included in the softkey labels or if the choices need more descriptive text than can fit into the softkey zones.

The list query displays the alternative choices in a selection list. The user can scroll the list and select an appropriate choice. The list query displays a separate prompt text above the list. The maximum prompt text takes up two rows, thus it should be simple and short. Like the confirmation query, the list query prompt text can contain the name of the object that the list query concerns.

When the list query is displayed, the highlight is on the first selection list item. Selections are done with the MSK, and the user can backstep from the list query with the RSK.

Design guideline: The list query should not be used if the choices in the list are only “Yes” and “No.” Confirmation queries can display more text in the prompt and consequently are more efficient for simple alternatives that only confirm or decline the query.

8.10 Soft notifications



Figure 22: Example of a soft notification

Typically, a soft notification reminds the user of an event that has occurred in the user's absence or while the user was busy with an application. Soft notifications are only displayed in the idle state.

The soft notification text in the main window can be static or dynamic. An animated icon may appear.

Soft notifications include, for example:

- Notification of all types of received messages (SMS, MMS, voice mail, e-mail);
- Notification of received data, such as business cards, calendar notes, and ring tones.

A soft notification has a context-specific function softkey to operate with the soft notification. If several functions are needed, the LSK is labeled Options.

The RSK is Exit by default. Pressing it discards the soft notification. After a soft-notification-related operation has been completed, backstepping or a long press of the End key returns to the idle state.

Soft notifications are stacked in case there is more than one pending at a time. After the topmost one has been proceeded with or discarded, the next one is displayed. Each soft notification has a priority value that determines the order of the notifications.

A soft notification can include a list of events. The list is used when several events have occurred and they are combined into one soft notification. The user can see more events at once and choose which event is opened and handled first. For example, if the user has received both a message and a missed call, the phone can display those two events at the same time. The user can scroll the list and select the one to open. The soft notification with list works in the same way as list queries; see Section 8.9, "Confirmation and list queries."

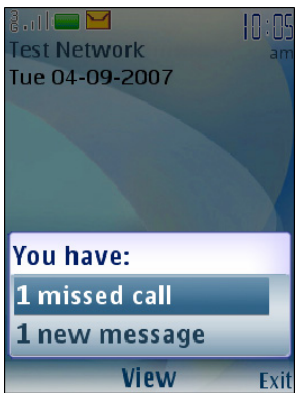


Figure 23: Soft notification with a list (combining communication events that have occurred in user's absence)

8.11 Notes



Figure 24: Example of a note

Notes inform the user of the current activities of the device, network, or mobile service, without requiring any user action, with the exception of a user response note. There are several types of notes.

- Wait notes inform the user to wait until an operation is completed. As an exception to other note types, wait notes may provide an option to the user to interrupt the operation. In that case, the RSK is labeled Quit. The RSK is labeled Cancel if the user is able to cancel the entire operation, for example, “Copying contacts” (in the Contacts application).
- Error notes inform the user about an error situation, for example, “Wrong PIN code.” Error notes should be avoided by creating UI designs that minimize the possibility of user errors.
- Warning notes inform the user about a situation that may cause harm to the user but that is not directly related to user actions, for example, “Battery low.”
- Information notes inform the user about the state of the device, network, or application, for example, “Memory full.”
- Confirmation notes notify the user of successful actions if there is no other indication about the accomplished operation, for example, “Time is set.”
- Permanent notes appear indefinitely and cannot be interrupted by the user with key presses. They stay on the display until the device dismisses them, for example, “Insert SIM card.”
- A user response note is an exception to other notes because dismissing it requires user action. A user response note is dismissed by the OK softkey.

Design guideline:

Notes typically disappear automatically after a timeout, except for the user response note, which requires user confirmation.

Notes can display with an icon (also, animated). A progress animation indicates that an action has started and is still progressing.

The entire note should display at once, except for the user response note, which is scrollable.

There should not be a sequence of notes, with the exception that after a wait note another note may display after the wait note has cleared.

9 Terms and abbreviations

Term or abbreviation	Meaning
Battery bar	Graphical element indicating remaining battery capacity.
Confirmation query	Requests the user to confirm an operation.
Cursor	Vertical element in editors indicating the insertion point.
Data query	UI component used for entering information, for example, a name or a phone number.
Fetch service	Perfoms retrieval of data from Contacts to be used in features.
Focus	Graphical indication of the current position in a list or grid.
Form	UI component used for presenting and editing several data elements, which are grouped together in the same layout.
Global exit	Exit to the device's idle state from any state by long-pressing the End key.
Header text	Textual aid, positioned at the top of the display, indicating to the user the current location in the application or menu structure.
Idle state	The basic state of the device where the device is waiting for input from the user or the network. The device goes to the idle state after startup.
Index number	Number, displayed in the status zone, indicating the position of the currently focused choice item in lists and grids.
Key repeat	The key's function is performed consecutively while the key is kept pressed down.
Keyguard	Feature that allows the user to lock the keys.
Level control	UI component that enables the user to adjust a level, such as volume, display contrast, or game level.
List query	UI component that requests the user to choose between alternative choices in order to proceed with an operation.
LSK	Left softkey. A dynamic key whose function is indicated to the user as a softkey label at the lower-left corner of the display.
Main zone	Area in the display that is used to display the content of the current application, for example, user-selectable items, user data, and message contents.
MMS	Multimedia Messaging Service. MMS combines conventional text messages with richer content types, such as photographs, images, voice clips, and video clips. Used with multimedia terminals, for example, WAP clients, which can receive and process multimedia messages.
MSK	Middle softkey. A dynamic key whose function is indicated to the user as a softkey label in the middle at the bottom of the display. Only in a three-softkey UI.
Multiselection list	Selection list that allows the user to scroll and select several (or no) options to be active.
Multi-tap input	Input method requiring many keystrokes per character due to the

Term or abbreviation	Meaning
	multiple mappings of characters to each numeric key.
Note	UI component that informs the user about the state of the mobile device.
Numeric keys	0-9, *, #
Pop-up	Pop-up windows are displayed on the top of the original display. The window is only as large as needed to display the information that it contains, so parts of the original display are visible behind the pop-up window. This allows users to see the context of the item that they are setting.
Prediction input	Input method that requires only one key press per character. Possible with an intelligent dictionary that allows the editor to predict what word the user intends to enter based on the number and combination of key presses.
RSK	Right softkey. A dynamic key whose function is indicated to the user as a softkey label at the lower-right corner of the display.
Scroll bar	Vertical bar with an indicator element, a thumb, at the right edge of the display that assists the user in navigation by expressing what portion of the data is visible on the display.
Scroll key	Hard key used for moving the highlight or cursor on the display.
Selection list	Screen that shows a list of choice items from which the user can select.
Signal bar	Graphical element indicating the network signal strength.
SK	Softkey. See *Softkey*.
SMS	Short Message Service. SMS is defined within the GSM standard, and it enables the point-to-point transmission of short messages. By means of concatenating messages, SMS can be used to transmit up to 31 KB of binary text or data. This service is currently available only as a mobile phone service.
Soft indicator	Textual indicator displayed in the idle state. May also contain a graphical icon.
Soft notification	Reminder of an event that has occurred in the user's absence or while the user was busy with an application. Soft notifications are displayed in the idle state.
Softkey	A dynamic, context-sensitive function displayed in close proximity to its physical key.
Softkey zone	Area at the bottom of the display that is used to display the softkey labels. Divided into two or three parts depending on the number of softkeys.
Status indicator	Graphical icon displayed in the status zone that is used to inform the user about device status or give other important information that should be easy to notice.
Status zone	Area at the top of the display that is used to display the header text, index numbers, and status indicators.
Tabs	UI component that enables fast viewing of parallel data views. The user can scroll between tabs to view different data presented in the tab

Term or abbreviation	Meaning
	pages.
Ticker zone	Area above the main zone that can be used to display a continuously scrolling text in MIDP 2.0.
UI	User Interface.
UI component	A building block of the user interface that is used to build an application or feature functionality and appearance.
User response note	A note that requires user confirmation.
Virtual button	A graphical element on the display that generates a function when it is focused and “pressed.”
Wait note	A note that informs the user that an operation is in progress.

10 Evaluate this resource

Please spare a moment to help us improve documentation quality and recognize the resources you find most valuable, by rating this resource.