S60 UI Style Guide

Version 1.2
November 9, 2005

S60 platform

Legal Notice

Copyright © 2005 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. About This Document 8

1.1 Purpose 8

1.2 Audience 8

2. Where the S60 User Interface Belongs 9

3. Hardware Requirements 10

3.1 Display 10

3.2 Keys 10

3.2.1 Extra keys 12

3.2.2 Full QWERTY keyboard 12

3.3 Memory Card 13

4. Graphical Components 14

4.1 Windows and Panes 14

4.1.1 Standard panes in application windows 15

4.1.2 Layout changes in right-to-left languages 20

4.1.3 Pop-up windows 20

4.2 Presentation of Text 21

4.2.1 Alignment 21

4.2.2 Truncation 21

4.2.3 Marquee scrolling 22

4.3 UI Themes 22

4.3.1 Replacing UI elements 22

4.3.2 Designing UI themes 23

4.3.3 Color scheme in a UI theme 23

5. Interaction Style 24

5.1 Keypad Functions 24

5.1.1 Key presses 24

5.1.2 Keypad tone 24

5.1.3 Typical functions of the standard keys 25

5.1.4 Typical full QWERTY keyboard functions 28

5.2 Navigation 29

5.2.1 Navigating in applications 29

5.2.2 Navigation using tabs 29

5.2.3 Hierarchical application structures 30

5.2.4 Navigation using links 31

6. UI Components 32

6.1 Lists and Grids 32

6.1.1 Highlighting 32

6.1.2 Empty lists and grids 32
6.1.3 List browsing 33
6.1.4 Order of items and browsing in grids 33
6.2 List Types 35
6.2.1 Menu list 36
6.2.2 Selection list 37
6.2.3 Markable list 39
6.2.4 Multi-selection list 41
6.2.5 Setting lists 42
6.2.6 Forms 45
6.3 List Layouts 48
6.3.1 Column structure of lists 48
6.3.2 List item types 50
6.3.3 Numbered items 53
6.3.4 Item type combinations 54
6.4 Grid Types 54
6.5 Find Pane 55
6.6 Options Menu 55
6.6.1 Submenus 56
6.6.2 Submenu indicators 57
6.6.3 Unavailable items 57
6.6.4 Other Options menus 58
6.6.5 Options template 58
6.7 Notes 61
6.8 Soft Notifications 62
6.8.1 Discarding soft notifications 64
6.8.2 Many simultaneous soft notifications 64
6.8.3 Information pop-up 64
6.9 Queries 64
6.9.1 Confirmation query 65
6.9.2 List query 65
6.9.3 Multi-selection List query 66
6.9.4 Data query 67
6.9.5 Message query 68
6.10 Indicators 68
6.10.1 Signal and battery indicators 69
6.10.2 Status indicators 69
6.10.3 Indicators in the navi pane 70
6.10.4 Editing indicators in pop-up windows 70

6.10.5 Soft indicators 70
6.10.6 Operator indicator 71
6.10.7 Application-specific indicators 71
7. S60 UI with Side Softkeys 72
7.1 Introduction to the Side Softkey Style 72
7.1.1 What is S60 UI with side softkeys 72
7.1.2 Default layout description and the basic principles of rotation 73
7.2 Interaction 74
7.2.1 Rotation 74
7.2.2 Basic interaction style 76
7.3 Designing Application UI for the Layout 77
7.3.1 Main pane 77
7.3.2 Status pane and the context pane 78
7.3.3 Title pane 78
7.3.4 Navi pane 79
7.3.5 Signal pane 79
7.3.6 Battery pane 79
7.3.7 Universal indicators 79
7.3.8 Control pane vs. softkey panes 79
8. Applications and Design Examples 81
8.1 Idle State 81
8.1.1 Active Idle 81
8.2 Application Shell 82
8.2.1 Shortcuts 83
8.2.2 Customizing Application Shell 83
8.2.3 Fast Application Swapping 83
8.3 Application Handling 84
8.3.1 Opening and closing applications 84
8.3.2 Multitasking 84
8.3.3 Application interactions 85
8.4 Phone 86
8.4.1 Call handling 88
8.4.2 Volume control 88
8.5 Viewers and Players 89
8.5.1 Text viewers 89
8.5.2 Image viewers 89
8.5.3 Multimedia viewers 89
8.5.4 Audio and video players 90
8.6 Text Editing 90

8.6.1 Editing menu 91
8.6.2 Editing indicators 92
8.6.3 Selecting text 92
8.7 Document Handling 93
8.7.1 Creating new documents 93
8.7.2 Saving edited data 93
8.7.3 Folders 94
8.7.4 Groups 95
8.7.5 Fetching data 95
8.7.6 Memory card in data handling 95
8.7.7 Embedded links 95
8.8 Settings 96
8.9 Help 96
9. Evaluate This Document 98

Change History

October 21, 2003	Version 1.0	Replaces documents *Nokia Series 60 UI Style Guide* and *Developer's Guide to Series 60 UI Category.*
February 25, 2005	Version 1.1	Document updated for Series 60 Platform 2nd Edition, Feature Pack 1 and 3.
November 9, 2005	Version 1.2	Chapter 7 added; small changes to cover S60 3rd Edition.

1. About This Document

1.1 Purpose

This document gives an overview of the S60 user interface and describes its essential parts, giving examples of how to use the interface elements.

The document can be used as an introduction to the style or as reference material. It can provide background material to help user interface designers make decisions about their products.

The style guide should give an overview and provide enough guidelines for designing good applications, but not all the information to write the software. This is intended to be a compact and easy-to-read guide, which means leaving out many details that can be found in other documents. There is no general discussion about good usability; instead this document tries to clarify how the style elements of the S60 user interface are meant to be used in practice.

The content is also intended to be independent of product-specific hardware, so that the guidelines would apply to any product that implements the user interface style. Sometimes this means leaving out information that would be appropriate for one product but maybe not for another one.

This document is based on and replaces the Forum Nokia (http://www.forum.nokia.com/) document *Developer's Guide to Series 60 UI Category*.

1.2 Audience

This document is intended, in the first place, for people who work with application design for S60 platform.

It will also help persons such as product managers to get a general view of the S60 user interface, what it is and how it is intended to be used.

2. Where the S60 User Interface Belongs

The S60 user interface is intended for use in mobile phones featuring personal information management (PIM) and multimedia applications such as:

- Calendars
- Text, multimedia and e-mail messaging
- Mobile browsers
- Imaging and audio applications and features

One hand operation is a key rule: the user is able to perform almost all tasks with just one hand, pressing the keys with the thumb. A few exceptions exist in functions that are targeted to power users and require pressing two keys simultaneously.

3. Hardware Requirements

The S60 user interface has certain requirements concerning the hardware. This section lists the assumed hardware for the concurrent implementations; it is possible to extend and modify the hardware to some extent.

3.1 Display



Figure 1: S60 user interface display

The S60 user interface display specifications are as follows:

- Resolution: the minimum of 176 pixels (width) by 208 pixels (height).
- Square pixels.
- Color capability (4096 or more colors preferred).

3.2 Keys

The following keys are required for the S60 user interface:

Navigation keys	Arrow up
	Arrow down
	Arrow left
	Arrow right
	The Selection key
Softkeys	The left softkey
	The right softkey
Call handling keys	The Call creation key
	The Call termination key
Other keys	The Applications key
	Alphanumeric keypad, with ITU-T numeric keypad (0-9, *, #) as default.
	The Clear key
	The Edit key
	The Power key



Figure 2: Default keys in the S60 user interface style

- The navigation keys can be ordinary buttons, or they can be implemented using different control devices, for example a 4-way pointing device or a roller which can be rotated and pressed so that the Arrow up, Arrow down and Select functions are mapped into it.

- The hardware solution may have some effect on the navigation functionality: for example, a long key press event cannot be accomplished with a rotating device.
- Both the softkeys have corresponding textual labels on the bottom of the screen.
- The Edit key is the only key that can be used simultaneously with another key press, for example, combinations where one of the navigation keys is pressed while the Edit key is held down.

See Section 5.1 for more information about the usage of keys.

3.2.1 Extra keys

Other specific keys can be added to a product to emphasize or facilitate some functions. These extra keys could be used to control applications or hardware such as spoken commands, sound recording, and audio volume control.

Camera key	Used for turning Camera on and for taking pictures.
Voice key	Used for voice dialing.
Volume keys	If a device has dedicated volume keys, they replace the scrolling keys' functionality in adjusting volume. A short press on a volume key adjusts the volume by one level, and a long press performs a continuous operation. Adjusting volume with the dedicated volume keys affects: • Applications that are in the foreground (and use volume). • Those applications (such as a player) that are active in the background. • Active applications in the background regardless whether the foreground application uses volume or not. The keys do not affect applications that are inactive in the background.

3.2.2 Full QWERTY keyboard

S60 platform also supports devices with a full QWERTY keyboard (from S60 3rd Edition onwards). See Section 5.1.4 for details on the keyboard usage.



Figure 3: Example of a S60 full QWERTY keyboard layout

3.3 Memory Card

A memory card can be added to a product to increase the available memory.

4. Graphical Components

4.1 Windows and Panes

The display layouts are hierarchically organized. The layouts are built using components called windows and panes.

Screen	The screen is the topmost display component, corresponding to the entire pixel area of the physical screen.
Window	A window is a component that has no parent except the screen. Typically, a window fills up the entire screen, but there are also smaller (temporary) windows that take up only a part of the screen, leaving other parts of the screen visible around themselves. Each application runs in a window of its own. Applications can also use other temporary windows.
Pane	A pane is a sub-component of a window. A window may contain several panes, and each pane may contain further sub-panes and so on. A bottom level component that cannot have a sub-component can be called an element.
Application window	An application window is a principal window filling up the entire screen. It is usually not used directly for display, but just as a parent for the various panes. A typical application window is divided into the following panes: • Status pane • Main pane • Control pane See Sections 4.1.1.1, 4.1.1.2 and 4.1.1.8 for more detailed descriptions on each of these panes.
Pop-up window	A pop-up window does not fill the entire screen; the pop-up window has a frame, and typically the underlying application is partly visible around the pop-up window. Pop-up windows are typically used in temporary states. Back stepping should not usually lead to a pop-up window. Detailed information on the various pop-up windows can be found in Section 4.1.3.

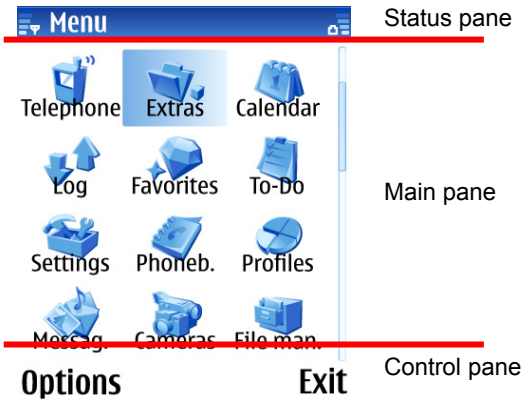


Figure 4: Panes

4.1.1 Standard panes in application windows

4.1.1.1 Main pane

The main pane is the principal area of the screen where an application can display its data.

There are a number of standard components for applications to use in the main pane:

List	There are several standard list types to choose from (see Section 6.1 for a detailed description of the list types.)
Grid	There are also several different grid types to choose from.
Find pane	The Find pane is used together with a list, and it allows the user to search list items alphabetically (see Section 6.1 for more information about the Find pane.)
Status indicators	The status indicator pane only exists in the Idle state, immediately below the navi pane, and displays status indicators. A few of the indicators appear also in the universal indicator pane (at other times than in the Idle state), others exist only in the status indicator pane.
Soft indicators	Soft indicators only exist in the Idle state. See Section 6.10 for more information on status indicators and soft indicators.

Applications can also use the main pane area to freely draw whatever is needed. In that case, however, the responsibility of the look and feel is entirely on the application's designer. General guidelines for designing application-specific main pane layouts can be found in Chapter 6.

4.1.1.2 Status pane

The status pane displays status information of the current application and state, as well as general information about the device status, e.g. the signal strength and battery charging. It occupies the top part of the screen.

The status pane may be non-existent in certain applications or situations.

The status pane contains the following sub-panes:

- Title pane
- Context pane
- Navi pane
- Signal pane
- Battery pane / universal indicator pane



Figure 5: Status pane areas

See the following subsections for detailed descriptions about each of these sub-panes.

4.1.1.3 Title pane

The title pane displays a context-dependent application title or state name.



Figure 6: Title pane

- Typically, the title text is the descriptive name of the current main pane view.
- In an application Idle state, the title text is typically the application's name.

4.1.1.4 Context pane

The context pane displays the current application's icon.



Figure 7: Context pane

- The user can recognize the application by the context pane whenever the title pane contains a context-specific text.
- The context pane graphic may contain some dynamic element (for example, an animation) to inform the user about the application's state. Certain applications (e.g. Contacts) may even use the pane in a fully dynamic way to display certain data relevant to the application. However, this kind of dynamic behavior should be minimized to avoid the user interface becoming restless

and ambiguous: the context pane is the main indication describing where the user is.

4.1.1.5 Navi pane

The principal use of the navi pane is to display information about the current state and view, and to help the user navigate in the application.



Figure 8: Navi pane

Depending on the context, the navi pane can alternatively contain:

Tabs	Tabs are used when there are parallel data views that can be viewed alternatively. There are arrow indicators at both ends of the widget (shown only when there are further tabs to scroll in the corresponding direction). Each tab has either a graphic or a text (or both) as a label. The following tab layouts can be used: • Two tabs • Three tabs • Four tabs Three long tabs (stacked so that only one is fully visible at a time) The currently active tab is highlighted. More than four tabs can exist simultaneously; the tabs can be scrolled horizontally. However, as a design guideline, the number of tabs should be kept small (the maximum of six tabs is recommended), and the number should not be dynamic (see Section 5.2.2 for a description of the effect of tabs on the navigation within an application).
Navigation text	A navigation text is displayed in the navi pane when there are similar items to be browsed by scrolling horizontally, e.g. dates in Calendar. Arrow indicators at both ends of the pane indicate the possibility to scroll.
Indicators	In editors, the navi pane contains editing indicators (see Section 6.10 for a more detailed description of the navi pane indicators.)
Application-specific content	When none of the above content types is suitable, the navi pane content can be designed specifically for an application (for example, volume slider).

Empty pane	The navi pane can be empty.



Figure 9: Tabs



Figure 10: Navigation text



Figure 11: Indicators in the navi pane



Figure 12: Empty navi pane

4.1.1.6 Signal pane

The signal pane displays the cellular signal strength indicator.



Figure 13: Signal pane (left)

The indicator may also contain information about a Packet Data (e.g. GPRS) connection status.

4.1.1.7 Battery pane / universal indicator pane

This area of the status pane is used in two different ways.

Battery pane	Displays the remaining energy level of the battery, using a graphical indicator. It also acts as a charging indicator.
Universal indicator pane	Used for displaying universal status indicators, i.e. status indicators that need to be visible regardless of the current application. The maximum number of items at a time is three; the items are prioritized according to their importance.



Figure 14: Battery pane (right)



Figure 15: Universal indicator pane (right)

4.1.1.8 Control pane

The control pane occupies the bottom part of the screen and displays the labels associated with the two softkeys.

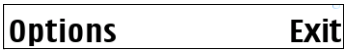


Figure 16: Control pane

The control pane is also active during **Options** menus, queries and other states using pop-up windows, although it does not reside itself in a pop-up window.

The actual physical softkeys should be positioned directly beneath the screen so that the association between the keys and their respective labels is evident.

4.1.1.9 Scroll pane

With any list, grid or other component that can be scrolled vertically, a scroll bar (introduced in S60 2nd Edition, Feature Pack 3) appears on the right hand side of the component. The scroll bar is displayed also on pop-up components.

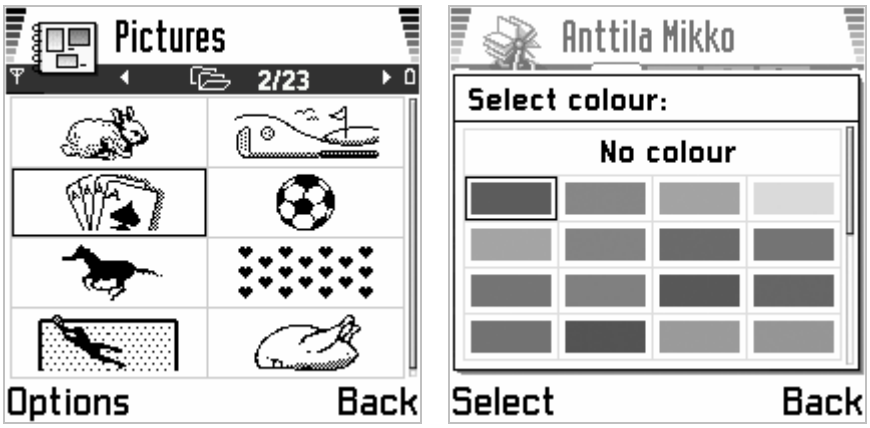


Figure 17: Scroll bar in the main pane grid (left) and in a pop-up window (right)

- The placement of the scroll handle on the scroll bar reflects the position of the focus on the scrolling content.
- The size of the scroll handle reflects the number of displayed items relative to the total number of items on the scrolled component.
- The scroll bar is displayed even when all items on the component can be displayed in the same view without scrolling. When this is the case, the scroll handle is displayed at the maximum size.

	Note: Until S60 2nd Edition Feature Pack 2, instead of a scroll bar, the scrolling indicator arrows are used. The arrows appear in the control pane between the softkeys, when there is a list that can be scrolled.

4.1.2 Layout changes in right-to-left languages

If the display text language is following right-to-left writing direction, also the standard panes change their places, and content can be mirrored if needed. The general rules for layouts in right-to-left UI languages are:

1. The status pane components are mirrored: the context pane is on the right, signal and battery panes change their places, etc.
2. The main pane lists and grids are mirrored: the A column icons and texts are on the right rather than on the left, etc.
3. The control pane is **not** mirrored: the left softkey (typically **Options** or a positive selection) remains on the left, and the right softkey (typically a negative command such as **Back**, **Exit**, or **Cancel**) remains on the right.



Figure 18: Example of a right-to-left layout

4.1.3 Pop-up windows

Certain UI components are displayed within pop-up windows. A common characteristic for all these components is that they are temporary states. This means that typically back stepping from one state to the previous state does not lead into pop-up windows; they are skipped.

More information on these components can be found in Chapter 6.

Options menu	The commands and options that are available in the current context can be accessed via the **Options** menu. It is displayed as a list in a pop-up window.
Query	A query is a component where the software requests user input. All query components are displayed in pop-up windows. They consist of a prompt (possibly containing a graphical element) and some kind of an input component. Various types of queries exist: • A Confirmation query has either one or two possible input values, given using the softkeys.

	• A List query has a limited number of possible input values, and the user selects one from a list. • A multi-selection List query has a limited number of possible input values, the user can select zero, one or more of them in one pass. • A Data query contains an input field for a numeric or alphanumeric value that the user can edit.
Note	A note is a feedback component that informs the user about the current situation. Notes do not require user input. They contain a text and possibly a graphical element and their layout is similar to a Confirmation query. However, the softkey labels are typically non-existent.
Soft notification	Soft notifications are reminders that inform the user of events that have typically occurred during the user's absence. Soft notifications can only be seen in the Idle state, and the user can acknowledge them. There are two types of soft notifications; the layouts resemble those of Confirmation queries and List queries: • A single soft notification contains one notification. • A grouped soft notification contains a number of information items presented as a list, and a title text common to all of the items.
Call window	Incoming calls and outgoing calls are presented in pop-up windows. See Section 8.4.1 for more detailed information on call windows.

4.2 Presentation of Text

4.2.1 Alignment

The default text alignment in left-to-right languages is on the left. There are only a few exceptions to this in specific cases, e.g. soft indicators in the Idle state, which are right-aligned.

In right-to-left UI languages, the default text alignment is on the right.

4.2.2 Truncation

When a text does not fit into the view where it is displayed, it should be truncated. By default, texts are truncated from the end, and three dots (…) are displayed at the end of the truncated text as an indication. There are two exceptions to this main rule:

- Phone numbers are truncated from the beginning, because the first digits of a phone number are usually considered less important than the numbers at the end.
- This general rule may be overridden in certain detailed components, where the latter part of the text is more differentiating than the beginning. Such might be e.g. queries to select an email address.

4.2.3 Marquee scrolling

When a text does not fit to a list view, the list text can be horizontally scrolling. Marquee scrolling should be used for texts that are not generated by the phone and for which truncation is not desired for some reason. For example, a web address may not fit to one row of text in a list and it may be important to see the whole address.

Scrolling is done only when the item is highlighted and the text is scrolled in only one direction.

4.3 UI Themes

A UI theme is used to change the appearance of the user interface for the phone's applications. A phone with the S60 user interface may provide several UI themes of which the user can activate one theme at a time. When using a UI theme, the appearance of the user interface changes, but the functions available with the application remain the same.

A UI theme is a customized graphical user interface (GUI) that replaces the phone's default look and feel.

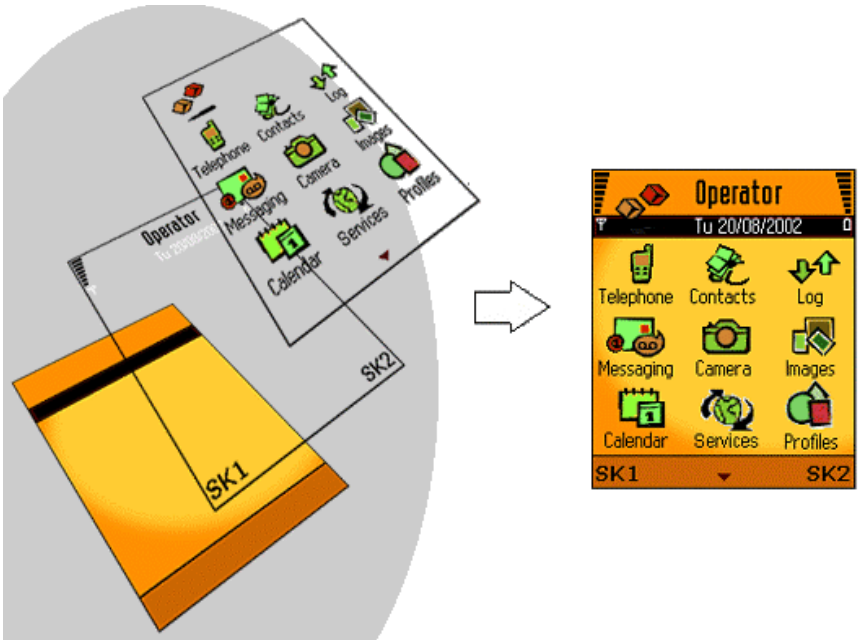


Figure 19: Simplified illustration of the UI theme concept

By activating a UI theme, the visual appearance of the user interface is changed. Note that this setting only affects visual elements, e.g. the interaction style, language settings or alert tones are not affected.

4.3.1 Replacing UI elements

To ensure a consistent look and feel in all applications, it is recommended to use the UI components that support UI themes. See the table below for examples of the core UI theme-supporting UI components:

Status pane:	• Signal pane (foreground

(Background bitmap for the whole status pane)	color/bitmap) • Battery pane (foreground color/bitmap) • Context pane (icon bitmap) • Title pane (foreground color) • Universal indicator pane (foreground color) • Navi pane (foreground color)
Main pane: (Background bitmap for the whole main pane)	• General list panes and grids • Find pane • Forms • Pop-up windows
Control pane:	• Background bitmap and foreground (text) color

In case an application uses UI components that a UI theme does not support, the factory default of the missing items will be used.

4.3.2 Designing UI themes

UI themes give the designer a freedom to design a unique GUI, and by designing bitmap backgrounds and icons, the result can be very different from the default settings.

A good GUI has consistent metaphor. When designing UI themes, a designer should make sure that the user interface supports the user's aim and does not mislead the user. It is also good to remember that even through the new look and feel offered by a UI theme would be fancy, the general user experience will fade if the performance of the terminal suffers because of a too large UI theme package file size.

A UI theme designer can decide which elements belong to a UI theme package. It is possible to design a UI theme that uses the default UI components of the phone together with components belonging to the UI theme.

4.3.3 Color scheme in a UI theme

Every UI theme can choose one color scheme from the system schemes.

5. Interaction Style

5.1 Keypad Functions

This section describes the typical functions for each key. Some application-specific functions may exist in addition to the ones mentioned here.

5.1.1 Key presses

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

Typically, the primary action of the key is performed when the key is pressed down, already before the key is released (there may be exceptions to this rule; see Section 5.1.3 for an example).

Some functions depend on the length of the key press:

- In a short press, the key is held down for less than 0.6 seconds (please note that this timeout value may depend on the product implementation). If the key is held down longer, the result is a long key press.
- Normally, if the pressed key (in the given context) has functions for both a short and long press, the short key press action is performed first at the moment when the key is pressed down, and if the key press turns out to be long, then the long key press action is performed. In the case of the Applications key and the Edit key, the interaction is different, causing the action on the key release event.
- Certain keys, possibly in certain contexts only, may perform key repeat. Key repeat starts after a long key press timeout when the key is continuously being held down, and the associated function is performed according to the key repeat frequency. For example, in scrolling within lists and text editors, these frequencies are by default 6 times/second for vertical and 10 times/second for horizontal scrolling (the repeat frequency may be product or application specific.)
- Long key press action and key repeat actions are not defined at the same time; only one can occur in the given context.
- The primary key press action should not be conflicting with the long key press action or key repeat action.
    - **Note:** With certain input hardware (e.g. Roller), long key presses and key repeat may not be possible. The long key press actions and key repeat actions should be designed so that this does not cause harm: the long key press must never be the only way to perform a function.

5.1.2 Keypad tone

A tone can be generated whenever a key event occurs. The tones for short key presses (actually a key down event) and long key presses are different; a key repeat event uses the long key press tone.

The keypad tone can be adjusted or turned off by the user.

5.1.3 Typical functions of the standard keys

Arrow up / Arrow down	• Moves the focus one item up/down in lists and grids. • Moves the cursor one line up/down in editors. • Scrolls the view up/down in viewers.
Arrow left / Arrow right	• Moves the focus one item to the left/right in grids. • Moves the cursor one character to the left/right in editors. • Moves to the previous/next view in tabbed views. • Moves to the previous/next document or view in certain document viewers. • Adjusts the sound volume during calls and sound playback when the device does not have dedicated volume keys. • Changes the value in a pop-up field immediately in forms.
Selection key	• Opens the focused item (e.g. a document or folder) in selection lists and grids. • Selects an option in menus and lists. • Opens the context specific Options menu when there is no item to open and no option to select (see Section 6.2.2). The Selection key must not directly activate any such function the user would not expect in the given situation: the Selection key is strictly a Get-in or Accept key. Therefore, the context specific **Options** menu shall be offered in states where no selectable items exist. The Open/Select function should not be mixed with the **Options** menu function within the same state; only one or the other should be used.
Left softkey	Typically labeled **Options**. Opens the **Options** menu. Other labels and functions: • **Select**. Used in menu lists and grids where further options are not available. Selects the focused item; same as the Selection key function. • **OK**, **Yes** and other positive replies; used in Confirmation queries. • In the Idle state, a shortcut to a specific

	application. Configurable by the user, labeled according to the application.
Right softkey	Typically labeled **Back**. Returns to the previous state. Other labels and functions: • **Exit** in applications' main states. • **Cancel**. Interrupts a procedure and returns to the previous state; used in queries and other temporary states. • **No** and other negative replies; used in Confirmation queries. • In the Idle state, a shortcut to a specific application. Configurable by the user, labeled according to the application.
Call creation	• Answers the incoming call when the phone rings. • Creates an outgoing call when in Phonebook and other states where the focus is in a field containing a phone number or a name associated with a phone number. • Sends a message; used when in a message editor and the **To** field contains a valid address. During calls: • Puts an active call on hold; activates a held call; swaps active and held calls if both exist. • Answers a waiting call (if only one call exists already) (see Section 8.4.1 for more detailed descriptions.) In the Idle state: • Brings up the Last Dialed Calls list for redialing.
Call termination	• Rejects an arriving call. • When there is an active call, ends the active call. • When there is only a held call, ends the held call. • When there are both an active and a held call, ends the active call and activates the held call. • When there are no calls and an application is active, returns to the Idle state (from S60 2nd Edition, Feature Pack 3 onwards the application is closed; in earlier S60 platform

	versions pressing the call termination key leaves the application running in the background). • A long press closes down all connections (for example GPRS, data call); however, this has no effect on IR and Bluetooth (see Section 8.4.1 for more detailed descriptions).
Applications key	• Opens Application Shell, allowing application launching and swapping. • When within Application Shell, returns to the Idle state. • A long press of the Applications key opens the Fast Application Swapping window, allowing switching between running applications.
Numeric keypad (0-9, *, #)	• Numeric and alphanumeric character entry. • Application-specific shortcuts and other functions.
Clear	• Clears characters when editing text or numbers. • Clears documents or other entities in lists and viewers (these functions always require a confirmation from the user). The Clear key is not used for back stepping or exiting; it is only used for deletion.
Edit	• Opens the editing menu in editors; the menu contains functions for input mode changing and other editing functions. Refer to Section 8.6 for the contents of the menu. • In editors, can be used together with the navigation keys to select (highlight) text, which then enables the **Copy** and **Cut** functions. • In markable lists, using the Edit key together with navigation keys allows the user to mark several items in the list, then a function can be executed on all the marked items as one operation. The Edit key is handled in a special way: the primary action (the editing menu) is opened from the key release event, not the key down event as usually. This is to enable the **Mark**/**Select** function where the key is being held down as a modifier key (see Sections 8.6 and 6.1 for more detailed descriptions of the **Select** and **Mark** functions.)
Power	• Turns the power on and off. This requires a long key press.

	Opens the list of Profiles for switching the active profile.
Volume keys	Adjusts the sound volume during calls and sound playback

5.1.4 Typical full QWERTY keyboard functions

The full QWERTY keyboard (an optional feature; supported from S60 3rd Edition onwards) default short key press events are the following:

Key	Action
Character key	Adds a character with key repeat.
Backspace	Moves the cursor back, clearing the previous character with key repeat.
Enter	Adds a carriage return where possible (a new line) with key repeat.
Space	Adds a space with key repeat.
Chr	Accented characters can be entered by holding down the Chr key and pressing a character key consecutively.
Scroll keys	Move the cursor of focus to the pressed direction with key repeat.
Character key	Adds a character with key repeat.

The full QWERTY keyboard key combinations are the following:

Key	Action
Shift + a character key	Enters the Shift-character of that particular character key. Can be used either as two consecutive short presses or simultaneous presses.
Shift + scroll keys	Selects text from the present cursor position to the direction of the selected scroll key. Simultaneous pressing required.
Chr + a character key	Accented characters can be entered by holding down the Chr key and pressing a character key consecutively.
“Dead” keys	Some keys in certain localized keyboards that produce a visible outcome in editing only after another key is pressed right afterwards. For example, accents over characters.
Ctrl + ‘x’	The **Cut** function in editing (for example, text that was selected with Shift + scroll key).
Ctrl + ‘c’	The **Copy** function in editing (for example, text that was selected with Shift + scroll key).

Ctrl + ‘v’	The **Paste** function in editing (for example, text that was selected with Shift + scroll key).

5.2 Navigation

The model of navigation is based on states arranged as hierarchical trees. The following added features bring in new flexibility:

- Tabs
- Applications key and Application Shell
- Direct navigation between sibling folders
- Links to applications and documents; Pinboard

These features are described in the following sections.

5.2.1 Navigating in applications

The traditional hierarchical tree structure forms the basis for navigation. The user can move forward from one node (state) by opening an available item or selecting an option from a menu. The **Back** function (available in the right softkey **Back**) returns to the previous level in the hierarchy. In the initial state of an application (number 1 in Figure 20), the **Exit** function replaces **Back** in the right softkey and is used for closing the application.

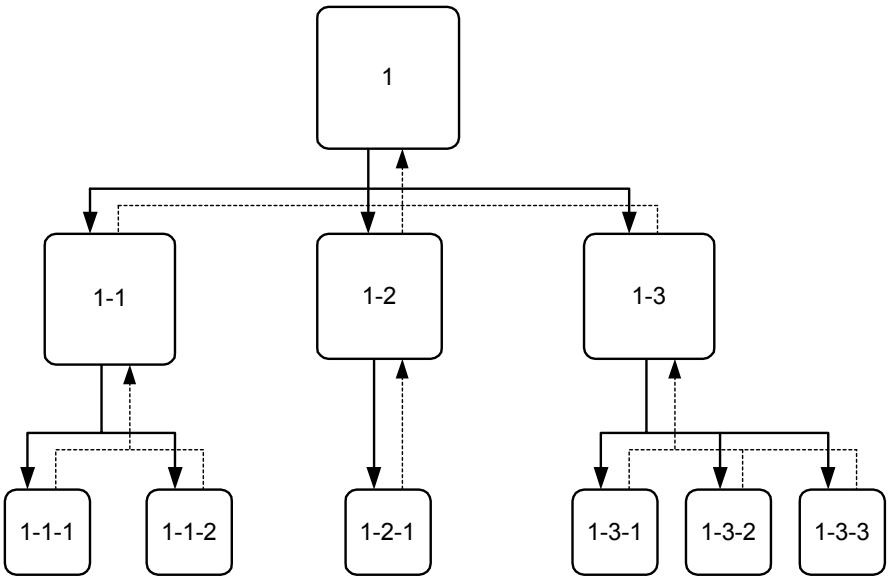


Figure 20: Example of the basic state hierarchy in an application. The solid lines indicate moving forward from a state into a sub-state. The dotted lines are backward moves to the previous level.

5.2.2 Navigation using tabs

The S60 user interface uses the tab metaphor that allows combining several pages of related information into a single state when all of it would not fit into a single screen or list. The user can switch the tabs using the Arrow left and Arrow right keys as indicated in the navi pane.

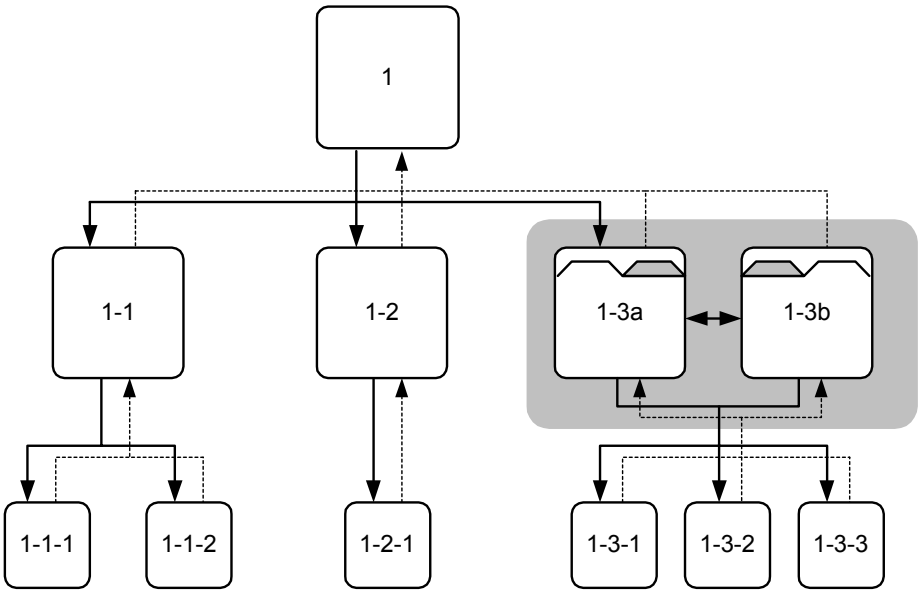


Figure 21: In this example, state 1-3 uses two tabs to present its information. The user moves between views 1-3a and 1-3b using the Arrow left and Arrow right keys. Note that there is no Back function moving between the tab views; Back from both of them leads to state 1.

Tab-controlled views apply the following rules:

- A list view from which each of the tabbed views can be accessed should precede the tabbed views one level higher.
- Moving from one tab view to another has no effect on the function of the **Back** softkey in these views: from all of them, the **Back** function leads to the same place, the previous level in the application. In this respect, tabbed views are interpreted as one state in the application.
- When a state has tabbed views, one of them is the default view that is opened when the user enters the state from the previous level.
- When the user has proceeded from a tabbed view into a deeper hierarchy level, the **Back** function returns to the same tabbed view where the user came from (which is not necessarily the default view described in the previous bullet).

The possibilities to go forward from tab views may differ from one view to another (although typically they are similar). This means that one tab view may have other functions available to the user than another tab view in the same state.

5.2.3 Hierarchical application structures

When browsing within a hierarchical application structure, the S60 user interface offers direct access from one folder to another (sibling) folder in the same level. The Arrow left and Arrow right keys are used to accomplish this. The navi pane displays the sibling folders as tabs.

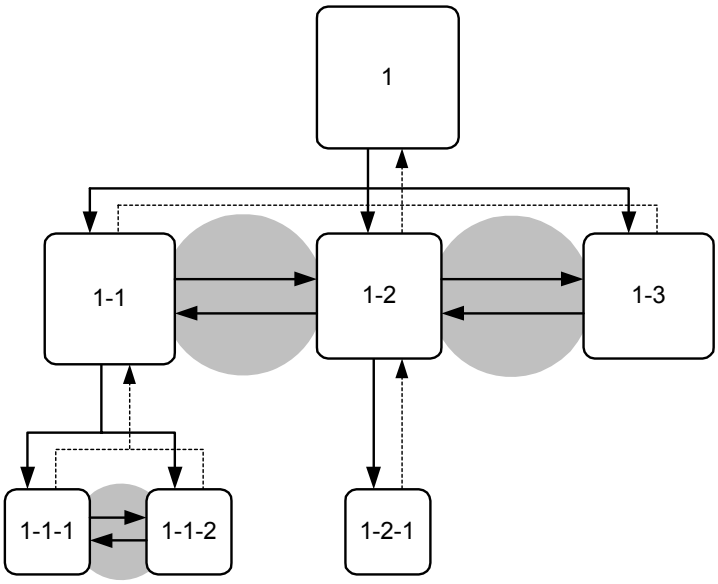


Figure 22: In this example, each displayed state is a folder containing a list of items. State 1 contains three folders (1-1, 1-2, 1-3). When the user has opened one of these, there is a possibility to move directly into the other two folders without first going back to state 1. Folder 1-1 contains two sub-folders and folder 1-2 contains one.

This additional navigation feature can be interpreted as a shortcut between sibling folders. For navigating hierarchical application structures, the basic navigation rules are applied, noticing the following:

- Moving from one folder to another has no effect on the function of the **Back** softkey in these views: from all of them, the **Back** function leads to the previous hierarchy level.
- The navigation shortcuts can only be applied when there is no other use for the Arrow left and Arrow right keys in the state.

5.2.4 Navigation using links

Links leading from one application (or the Idle state) to another application may exist. Links are one-way shortcuts: there is no direct path back to the state where the link was started; instead, the navigation inside the linked application functions as if the user had manually activated the other application and navigated to the target state.

6. UI Components

6.1 Lists and Grids

Vertical lists are used extensively in most applications. Two-dimensional grids are less frequently used, but have obvious advantages in some situations. In lists and grids, the user can move the focus from one item to another using the navigation keys. This is called browsing or scrolling.

In general, vertical browsing is preferred over horizontal browsing; the keypad solutions should take this into account.

6.1.1 Highlighting

When a list or grid is in use, one item on it is always in focus. The item in focus is indicated by a graphical means called highlighting.

The appearance of highlighting on a list item is a colored (or gray) bar which completely occupies the item in focus. The item text and graphics are displayed on the bar.

Grid item highlighting is a frame over the item in focus.

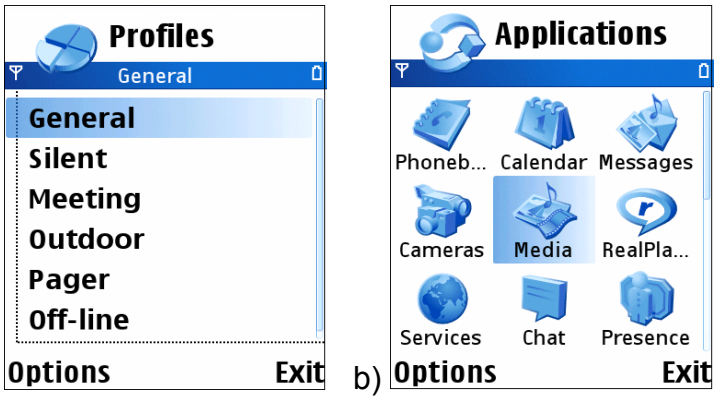


Figure 23: List highlighting (a) and grid highlighting (b)

6.1.2 Empty lists and grids

If there are no items to be displayed in a list or grid, the pane contains a text informing the user about the empty list.

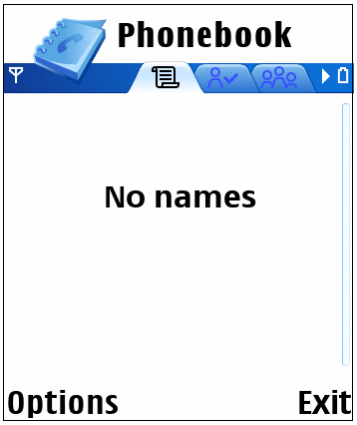


Figure 24: Empty list

Depending on the case, the text may be aligned to prompt the user to create the first item for an empty list.

6.1.3 List browsing

In a list, browsing is possible in two directions: pressing the Arrow up key moves the focus one step up (backward), and pressing the Arrow down key moves the focus one step down (forward).

When browsing, the item in focus must always be visible. The detailed rules of moving the focus are as follows:

- If the choice item that is becoming focused is already fully visible, all the items remain in their current positions, and the highlight is moved from the old item to the new item.
- If the new choice item is not visible, all items are moved in the view to the appropriate direction so that the new item becomes fully visible. For example, if the focus is moving down and the new item is currently below the bottom edge of the view, the items are moved up.
- When moving the choice items in the view, they are only moved the minimum amount necessary. For example, when moving items up to get the next one under the bottom edge visible, the item moves to the lowest allowed position in the view.

By default, all lists loop. This means that it is possible to browse forward from the last choice item, which leads to the first item in the list, and vice versa.

It is possible to specify that a list is a queue. This means that it is not allowed to browse forward from the last item or browse backward from the first item. If the user attempts this, the list does not react; there is no feedback except the normal keypad tone.

6.1.4 Order of items and browsing in grids

In a grid, the available items are in a rectangular arrangement of cells and browsing is possible in four directions. In addition to the up and down functions, the user can press the Arrow right key to move the focus one step to the right, or the Arrow left key to move the focus one step to the left.

The number of items can be larger than what fits in the view, so the grid items may scroll in the view when browsing.

- The preferred scrolling dimension is vertical; this means that when more items are added, the number of items in a grid grows downwards line by line, but not outside the window to the left or right.
- A grid should not be scrollable in both dimensions; it is acceptable only in cases where the grid has a natural geometry that cannot be changed. Calendar's Month view is an example of this kind of geometry (but even in that case it is better to fit the whole month on the screen rather than make it scrollable in both dimensions).
- The default filling order of choice items in a grid is first left–to–right, then top-to-bottom. In right-to-left languages, the order is flipped horizontally.

a.

1	2	3
4	5	6
7	8	9

b.

3	2	1
6	5	4
9	8	7

Figure 25: Default grid-filling order: a) Left-to-right UI languages b) Right-to-left UI languages

In certain cases, it is possible that a grid is not filled completely. Depending on the application, the grid can be auto-filling (for example, the cells are moved within the grid so that empty cells in the middle get filled), in which case there can only be empty cells on the rightmost part of the last line. Other applications may allow empty cells anywhere, so that the grid can be sparse.

Browsing in grids that scroll vertically resembles traditional scrolling in text editors, based on the idea that the user can always move to the correct row first and then move within the row to the correct item. The following rules are applied:

- Empty cells are skipped: the focus is never on an empty cell.
    - An exception to this occurs when the user is moving items around in a grid; in that case, all cells are accessible.
- When browsing up or down, the focus is moved to the adjacent cell directly below or above the current cell if that cell is filled. In case it is empty, the nearest cell towards the beginning of the same row gets the focus. If all cells on the row are empty, the search continues on the next row in the same direction, and so on, until a filled cell is found.
- When browsing towards the end of a row, the focus moves to the following filled cell on the same row. If there are no filled cells in that direction on the row, the search continues from the beginning of the next row, and so on, until a filled cell is found.
- Browsing towards the beginning of a row moves the focus to the previous filled cell on the same row, or continues searching from the end of the previous row. Using only the Arrow right or Arrow left key, the user can thus go through every item in the grid, regardless of the distribution of items in it.
    - **Note:** In right-to-left UI languages, such as Arabic and Hebrew, the end of a row is on the left-hand side. Respectively, the beginning of the line is on the right.

- The grid is scrolled (moved within the view) only when the item that is becoming focused is not fully visible already.
- A grid may also loop vertically within the same column. When browsing down from the cell at the bottom of a column, the focus moves to the choice item at the top of the next column and vise versa. If the focus is in the last column when applying the previous rule, the focus loops over to the top of the first column and vice versa.

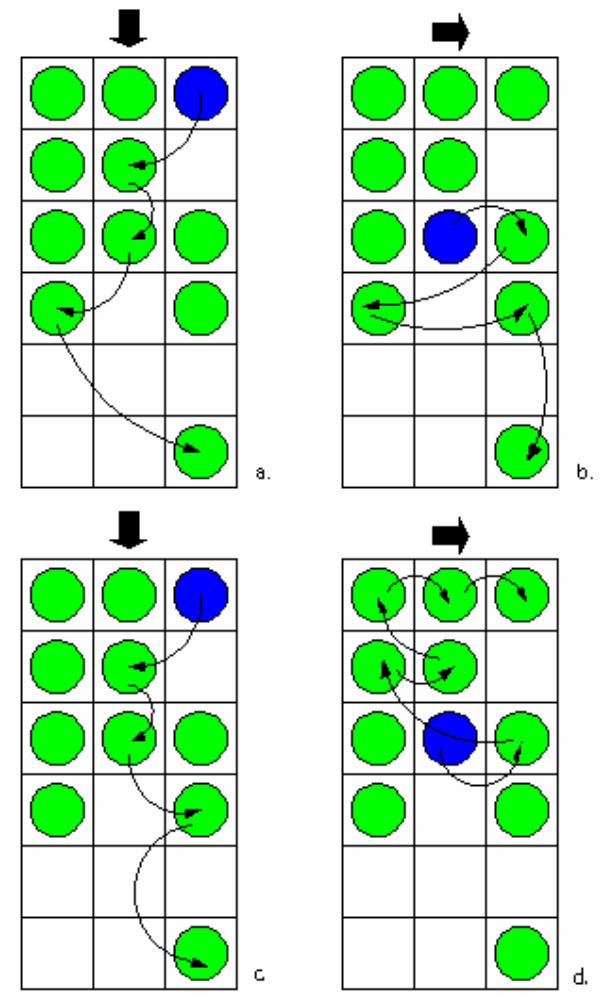


Figure 26: Examples of moving the focus in a grid.
a: Left-to-right UI language; only scroll down commands used
b: Left-to-right UI language; only scroll right commands used
c: Right-to-left UI language; only scroll down used
d: Right-to-left UI language; only scroll right used

6.2 List Types

Based on the functionality, the following list categories can be identified:

- Menu list
- Selection list
- Markable list

- Multi-selection list
- Setting list
- Form

Lists belonging to one category may have different graphical appearances. See Section 6.3 for information about different looking list items.

On the whole, list items and commands should not be mixed, but in cases where some command in a list would significantly boost efficiency and user experience, it is perfectly acceptable to add one command onto the list. It is important that there is no more than one command per list, and that the command is placed as the first item on that list.

6.2.1 Menu list

Menu lists are used to select one item from a list and do nothing else; the **Options** command is not available when a menu list is being browsed (the **Options** menu itself is a menu list).

If a high-priority event, such as an incoming call, occurs when a menu list is open, the list is cancelled and the new event takes control. These lists are often displayed inside a pop-up window.

The default keypad functions in a menu list:

Arrow up / down	Moves the focus in the list.
Arrow left / right	Ignored (unless there is a submenu; see Section 6.6).
Selection key	Selects the item, does the associated function.
Left softkey (**Select**)	Selects the item, does the associated function.
Right softkey (**Cancel**)	Dismisses the menu; returns to the state preceding the opening of the menu.
Call creation key, Edit key	Ignored.
Numeric keypad	Ignored.
Clear key	Ignored.
Other keys	Dismiss the menu and do the default action of the key.



Figure 27: Options menu

6.2.2 Selection list

A selection list is a common means of displaying and accessing data in applications. When a selection list is displayed, the application is typically in a permanent state, which means that the user can leave the application, open another application and later return to the same state. Selection lists are displayed in the main pane.

Typically, the user can open items in a selection list, leading into another, more detailed view of the item within the application. In addition to browsing and selecting items, other functions are available in the **Options** menu (see Section 6.6).

The usage of the Selection key in selection lists deserves special attention. Depending on the case, it can do the following actions:

- Selects the item in focus. This should happen whenever it is assumed that the user is clear about what happens. Selecting can mean:
  - Opening an item, such as a folder or a date in Calendar, leading to a detailed view.
  - Executing a command when the focused item is a command.
- Opens the context specific **Options** menu. This should only happen when the user cannot be assumed to know what happens if the Selection key is pressed. The menu should contain only high-priority options associated with the item in focus, not general items such as **Settings**, **Help**, or **Exit**. The maximum amount of options in the context specific **Options** menu is four.

The two types of Selection key actions should not be mixed within one list; one or the other should happen for every item in the list.

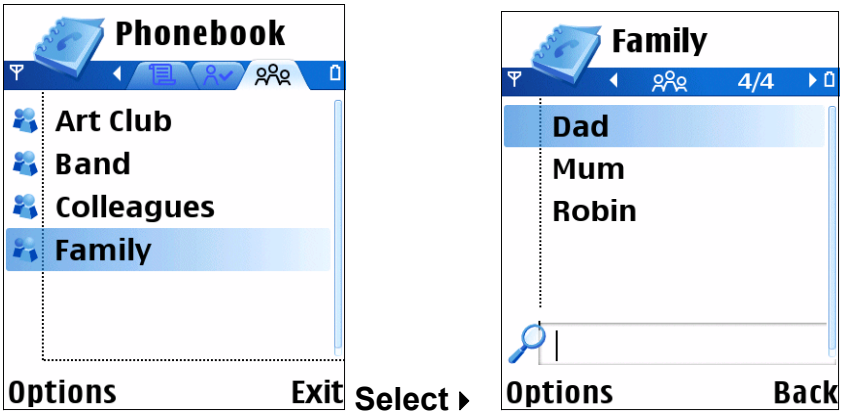


Figure 28: Select action opens a view

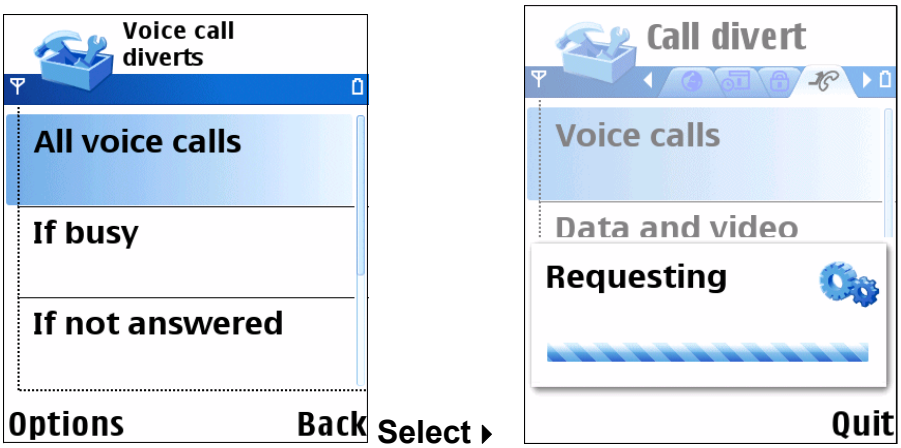


Figure 29: Select action performs a command

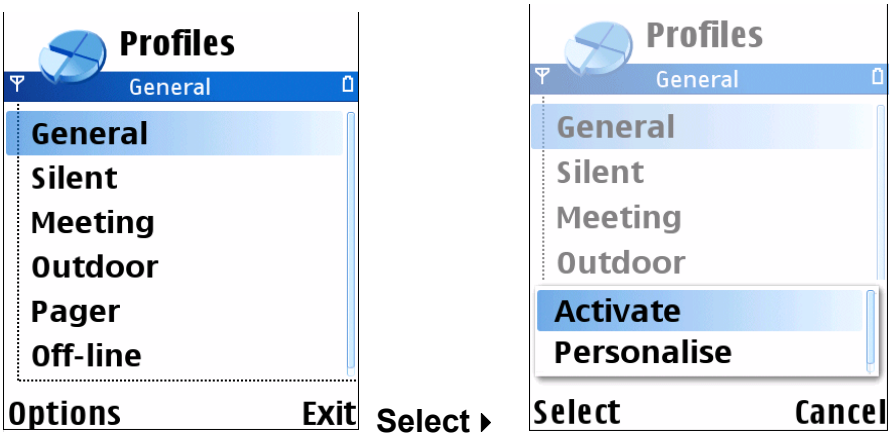


Figure 30: Context specific Options menu opened with the Selection key

The keypad functions for selection lists are as follows:

Arrow up / down	Moves the focus in the list.
Arrow left / right	May be ignored, or may have navigation functions associated with them (see Section 5.2).

Selection key	Selects the item; see the discussion above.
Left softkey (**Options**)	Opens the **Options** menu (see Section 6.6).
Right softkey (**Back**; **Exit**)	Back stepping (see Section 5.2).
Clear	Deletes the item if it can be deleted (confirmation from the user is required); otherwise ignored.
Edit	Ignored, or a marking function if the list is markable (see Section 6.2.3).
Numeric keypad	May be ignored, or may have specific functions within the state.
Other keys	Do the default action of the key.

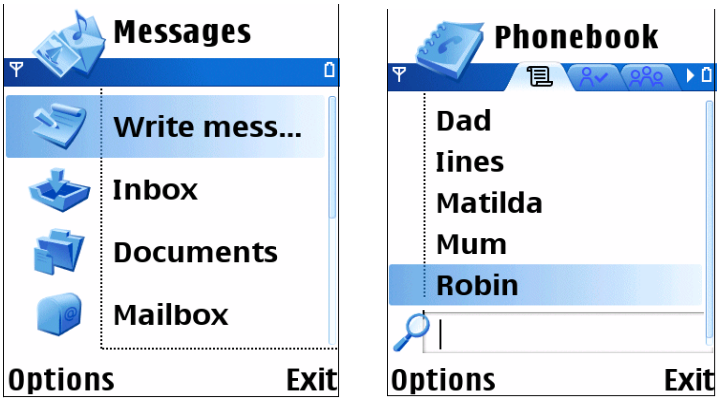


Figure 31: Selection lists

6.2.3 Markable list

A markable list is a selection list with the added marking feature. The user can mark any number of items on the list while browsing it, and then execute a single command, which is applied to all the marked items. This is analogous to the multiple-item highlight feature used in computer GUIs.

The marking feature is typically applied to lists the user manages, containing a large number of items that may be for example sent, moved or deleted. By marking items first and then selecting a command, the user can do certain operations more quickly and with less key presses than doing the command separately for each of the items.

Marked items are indicated graphically.

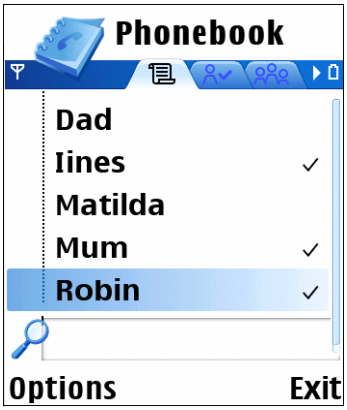


Figure 32: Markable list. Marked items are indicated graphically in the list.

The **Mark** and **Unmark** functions are available in the **Options** menu of the markable list. Alternatively, or as a shortcut, the user can keep the Edit key pressed while using the navigation keys and the Selection key in the following way:

- Pressing the Selection key while holding down the Edit key marks the current item. This is a toggling function, so pressing Edit-Select on a marked item unmarks the item.
- Pressing an Arrow key (up or down) while holding down the Edit key marks both the current item and the one onto which the focus moves. If the user keeps holding the Edit key and scrolls further in the same direction, all the scrolled items become marked; to unmark the items in reverse order the user can scroll into the opposite direction while holding the Edit key down.
- Items can be unmarked by Edit-scrolling: if the user starts holding the Edit key down when on a marked item and then scrolls, all scrolled items become unmarked.
- Several marking actions can be done subsequently. The user can mark an item, then release the Edit key, browse and move the focus onto some other item on the list, and then mark that item. The first item remains marked, and the ones between the two do not become marked.
- All items become unmarked when the user exits the list, for example by back stepping. There may be exceptions to this rule in certain applications that require selecting items from several levels of folder hierarchy. An example of this is creating a play list in the Media Player application. Items remain marked if the user opens and cancels the **Options** menu, or swaps applications.
- Pressing and releasing the Edit key alone does not cause any action.
- If the user presses the Selection key (without pressing the Edit key) when there are marked items on the list, the context specific **Options** menu is opened, containing only the functions that apply to multiple items.

A markable list functions exactly like a normal selection list, except for the marking feature.

The user may access the **Options** menu to perform functions on all marked items at once. Appropriate error handling must be designed for functions that do not apply to some or all of the marked items.

- When executing a function, if any of the items are marked, all the marked items are affected by the function. If the focus is on an item that is not marked, the function does not affect that item.
- After the selected function is successfully done, all items are unmarked. In an error case, when the function cannot be applied, the marks should remain in place.

The **Options** menu includes the marking and unmarking functions in a submenu so that any user can find the feature. The submenu has also the **Mark all** and **Unmark all** options.

For other keypad actions, see Section 6.2.2.

6.2.4 Multi-selection list

Multi-selection lists are used when the purpose is to emphasize that it is possible to select several items from a list at the same time. Typically, there is an operation going on that expects one or more items as input. An example of this is when the user is creating a group, and a list of names is offered as a multi-selection list.

In a multi-selection list, the user can browse the items and check and uncheck any number of them. The state of each item is indicated in a checkbox adjacent to the item. When the user accepts the list, information about the marked items is passed to the application.

Items are checked and unchecked using the Selection key, and the list is accepted with the left softkey **Done/OK**. Note that unlike in a markable list, the **Options** softkey is not available: one can only check and uncheck items and then accept or cancel the list.

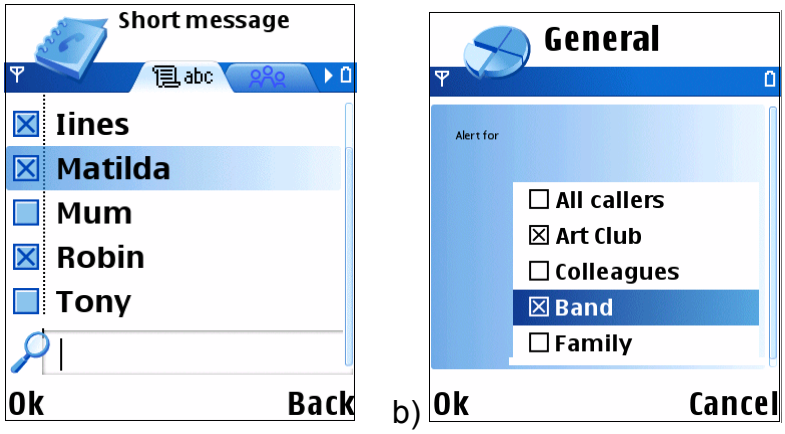


Figure 33: Multi-selection lists: in the main pane (a) and in a setting editor (b)

The keypad functions for multi-selection lists are as follows:

Arrow up / down	Moves the focus in the list.
Arrow left / right	Ignored (can be used to control tabs).
Selection key	Marks / unmarks the current item; toggle.
Left softkey (**Done**)	Accepts the list, passes the selections to the

	application.
Right softkey (**Cancel**)	Cancels the list, returns to the previous state.
Clear, Edit, Call creation	Ignored.
Numeric keypad	Ignored.
Other keys	Do the default action of the key.

6.2.5 Setting lists

A setting list is a specific kind of selection list containing setting items that the user can adjust. Setting lists are displayed in the main pane.

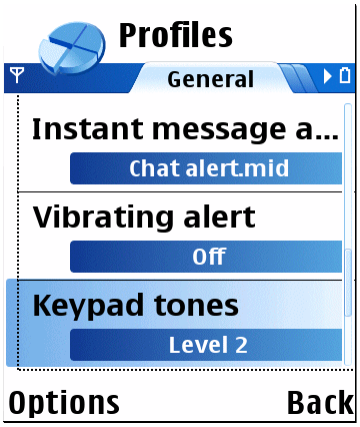


Figure 34: Setting list

A setting item can be adjusted by selecting it in the same way as selection list items are selected in general, i.e. by pressing the Selection key, or choosing the **Change** command from the **Options** menu. The main pane then displays the setting item editor where the value can be changed.

There are several setting item types available. They look the same in the setting list: each item displays an attribute text (the title of the setting) on one line and the current value on another line within the item. The adjusting and editing functions differ between the setting item types.

Pop-up setting	A pop-up setting allows the user to choose one value from a pre-defined list. The setting editor displays the available values in a menu list. A pop-up setting may also allow the user to enter a new textual value in addition to the pre-defined values. In that case, the last option is named **Other** (or equivalent), and selecting it opens a Data query for entering the new value.
Multi-selection list setting	A multi-selection list setting allows the user to choose several simultaneous values from a pre-defined list. The setting editor displays the available values as a multi-selection list. The setting item displays the number of selected items versus all the items in the value field; for example: 3/8.

Text setting	The value of a text setting item is an alphanumeric or numeric string. The editor can be of a specific type, such as the Date and Time editor.
Slider setting	With a slider, the user can adjust the value of a setting by sliding a marker. The value of a slider is adjusted using the Arrow left and Arrow right keys.



Figure 35: Pop-up setting



Figure 36: Multi-selection setting

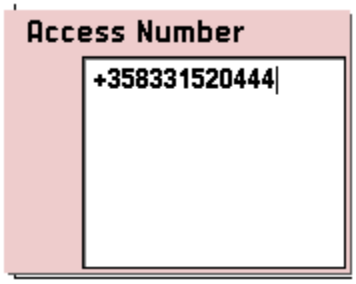


Figure 37: Text setting

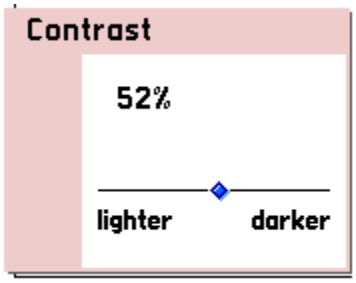


Figure 38: Slider setting

By default, in a setting editor, the keypad functions are as follows:

Arrow up / down	• Pop-up, multi-selection: browses the list. • Text: moves the cursor. • Slider: ignored.
Arrow left / right	• Pop-up: ignored. • Multi-selection: ignored. • Text: moves the cursor. • Slider: adjusts (a key repeat event may move the slider by several steps).
Selection key	• All except multi-selection: accepts the value, returns to the setting list. • Multi-selection: toggles **Mark**/**Unmark**.
Left softkey (**OK**)	Accepts the current item or value, returns. (**Note:** may also be **Options** in certain instances.)
Right softkey (**Cancel**)	Cancels the setting editor, returns.
Clear	• Pop-up, multi-selection: ignored. • Text: deletes characters. • Slider: ignored.
Edit	• Pop-up, multi-selection: ignored. • Text: opens the editing menu. • Slider: ignored.
Numeric keypad	• Pop-up, multi-selection: ignored. • Text: input functions. • Slider: ignored; if the value is numeric it may be adjustable using the Numeric keys.
Other keys	Do the default action of the key.

In the setting editor, the navi pane is empty, or in case of a text editor, contains the editing indicators.

Further guidelines:

- Typically, a setting list is accessed via the **Options** menu (see Section 6.6).

- Setting lists can only contain setting items, not other types of items mixed with them. (In case the settings are arranged in a hierarchical structure, both setting items and setting folders may exist in one list. See Section 8.8.)
- All the setting item types can co-exist in the same setting list.

If a pop-up setting has only two available values (such as **On** and **Off**) and no special procedure is needed when switching from one value to the other, then the setting editor (list) need not be displayed when the user presses the Selection key; the item's value is changed immediately. However, if the user opens the setting item via the **Options** menu, or if an additional procedure is required (for example a password) before the item's value can be changed, the list is opened normally.

6.2.6 Forms

A form is a specific kind of selection list, where all items (fields) have some editable content.

A form can be in the View state or in the Edit state. The item layouts and functionality are different in these states:

- In the View state, the items are not editable. The form functions and looks exactly like a similar selection list. Items can be selected to perform an application-specific function.
- In the Edit state, the user can edit all the fields. Forms can contain text fields (alphanumeric or numeric content), pop-up fields and sliders.

The user can switch from the View state to the Edit state using the **Edit** command in the **Options** menu.

In the Edit state, the contents of the form can be accepted using the right softkey, labeled as **Done**. The form then returns to the View state.

6.2.6.1 Always-editable forms

In case the View state is not useful, the form can be specified as Edit-state only. Then the user can edit the fields right away when entering the form, and accepting the form returns into a state outside the form instead of the View state.



Figure 39: Form in the Edit state

6.2.6.2 Forms without the Options softkey

In case a form does not need any context-specific functions in the **Options** menu, the softkey interface can be the same as in queries: the left softkey is **Done** and the right softkey is **Cancel**. **Done** accepts the contents and returns, whereas the **Cancel** function discards all changes in the form and returns.

6.2.6.3 Form items

Empty items (that contain no data) can be hidden in a form's View state. However, this is not a requirement; forms can be designed either way, whichever is better for the given application.

Unlike ordinary list items, the form items may have different sizes in the layout but in the Edit state only. A long data field may occupy more than one line in the Edit state, but in the View state it is truncated to the first line.

The user may be able to add and remove form items. This is done with commands in the **Options** menu.

Form items always have a label. The label has a text part, or a text and a graphical part; however all items in one form must have the same column layout (see Section 6.3 for detailed information about layouts).

The following item types can be used in a form:

- Text field (alphanumeric or numeric content)
- Pop-up field
- Slider

Any combination of these types is possible within one form.

When a form is in the Edit state, the user can move the focus up and down like in a list. The highlight in the Edit state is different from the highlight in the View state, acting as a visual cue. The cursor blinks in the text field that is in focus. There is no need to save each field separately; the user can browse and modify the fields in any order and then accept all modifications. During the browsing of a form in the Edit state, the keypad functions are as follows:

Arrow up / down	Move the focus between form items (when in a text field, move the cursor within an item line by line).
Arrow left / right	• In a pop-up field: change the value without opening the list. • In a text field: move the cursor character by character. • In a slider: adjust the slider value.
Selection key	• In a pop-up field: opens the list. • In a text field: opens the context specific **Options** menu or no action.
Left softkey (**Options**)	Opens the **Options** menu (in a form without the **Options** menu, the left softkey is **Done**).

Right softkey (**Done**)	Accepts the contents and returns to the previous state (in a form without the **Options** menu, the right softkey is **Cancel**.)
Clear	• Pop-up: ignored. • Text: deletes characters. • Slider: ignored.
Edit	• Pop-up: ignored. • Text: opens the editing menu; selects text. • Slider: ignored.
Numeric keypad	• Pop-up: ignored. • Text: input. • Slider: ignored.
Other keys	The default action of the key.

The field types are described in the following paragraphs:

Text field	A text field contains some type of numeric or alphanumeric data. It can be edited directly using the usual editing functions when the form is in the Edit state. The text field can expand to more than one lines if necessary. In the View state, a text field looks identical to a corresponding list item.
Pop-up field	A pop-up field offers the possibility to choose one value from a pre-defined list. In the View state, a pop-up field looks identical to a list item; the text is the current value of the field. In the Edit state, it has a distinct look that identifies the field as a pop-up list. To edit an item in the Edit state, the user can press the Selection key; it opens a menu list that contains the available values. The highlight is on the current value. When the list is open, the softkeys are **OK** and **Cancel**, as usual with a menu list. Both the softkeys return to the Edit state in the form. A pop-up field may also allow the user to enter a textual value in addition to the pre-defined values. In this case, the last option is named for example **Other**, and selecting it opens a Data query. See Figure 40 below.

Slider	With a slider, the user can adjust a numeric value (although it is not necessary to present the actual value to the user as a number). In the View state, the item is presented as a textual item. In the Edit state, the value of the slider is immediately adjustable using the Arrow left and Arrow right keys.

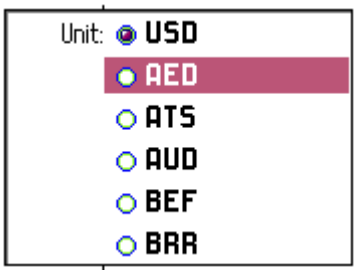


Figure 40: Pop-up field

6.3 List Layouts

List items can in general be more complex (contain more elements) than grid items. Certain layout rules apply to all lists:

- All items in a list have equal height on the screen (forms do not follow this rule in the Edit state, see Section 6.2.6.)
- The column structure of all items in a list must be similar. It is not possible to combine e.g. single-column items to a three-column item list (however, it is possible to use some different item types having the same general appearance), see Section 6.3.1 for columns.
- Partial items are not visible: when the list/grid pane area does not exactly correspond to an integer number of items, the remaining area outside the last fully visible item appears empty, displaying the background of the particular pane.

6.3.1 Column structure of lists

For visual consistency, the standard list layouts are built around a structure of three virtual columns. The column borders are aligned with the sides of the context pane.

The width of list items can be divided in three sections (columns A, B and C). All three columns need not be used separately in a list layout: combinations AB, BC or ABC are possible. However, all items within a list must use the same column layout.

Additional indicator icons can be displayed in the right edge of column C (sometimes called column D), as seen in example b) in Figure 41 below. The area for these indicators is not really a column in the same sense as the other three, as it can be used dynamically, item by item.

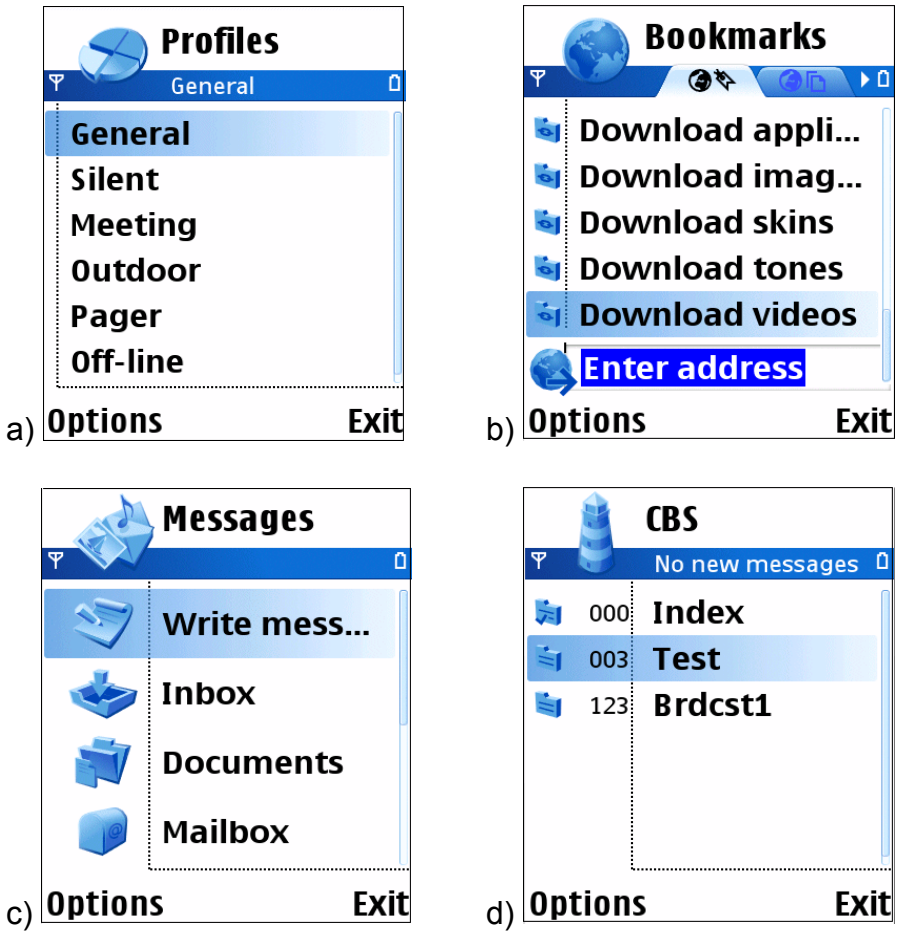


Figure 41: Different column arrangements:
a) All columns used for a text item
b) Column A contains an icon, columns B&C combined for text
c) Columns A&B combined for a large icon, column C contains text
d) Column A: an icon; column B: a label; column C: text

The standard elements in list items, associated with columns, are:

Column A	• Small graphic (icon) • Item number (see Section 6.3.3)
Column B	Heading (title or attribute of the item)
Column AB	• Heading (title or attribute of the item) • Large graphic (e.g. icon or image thumbnail)
Column C/BC/ABC	The main text of the item

Note: The list layout can be mirrored for localizations such as Arabic and Hebrew (for example, column A is in the right edge of the pane). See Section 4.1.2.

6.3.2 List item types

The appearance of list items can be chosen from the following types. The examples are from lists displayed in the main pane; for most item types a similar component for use in pop-up windows does exist.

Single-line item	Main text only (columns ABC). Usage: • Menu lists • Selection lists, markable lists There is a corresponding component for pop-up windows, without the line on the left side.
Single-line item with heading	Heading text (AB), main text (C). Usage: • Menu lists • Selection lists, markable lists • There is a corresponding component for pop-up windows.
Single-line item with graphic	Small graphic (A), main text (BC). Usage: • Menu lists • Selection lists, markable lists • Multi-selection lists There is a corresponding component for pop-up windows.
Single-line item with graphic and heading	Small graphic (A), heading text (B), main text (C). Usage: • Menu lists • Selection lists, markable lists • Multi-selection lists There is a corresponding component for pop-up windows.
Single-line item with large graphic	Large graphic (AB), main text (C). Usage: • Menu lists • Selection lists, markable lists

Two-line item	Main text (ABC). Usage: • Menu lists • Selection lists This layout has one text item that can extend to two lines.
Double item	Primary text, secondary text (ABC). Usage: • Menu lists • Selection lists This layout has two text lines (primary text on top, secondary text below). The second line may be empty. There is a corresponding component for pop-up windows.
Double item with graphic	Primary text, secondary text (ABC). Usage: • Menu lists • Selection lists
Double item with large graphic	Large graphic (AB), primary text, secondary text (C). Usage: • Menu lists • Selection lists Similar to a double item, but with the graphic added on the left side. The second line may be empty. There is a corresponding component for pop-up windows.
Double item style 2	Heading text (ABC), main text (BC). Usage: • Forms • Menu lists • Selection lists This layout has the emphasis on the lower line, as opposed to the other double item style where the upper line is emphasized.
Double item style 2 with graphic	Graphic (A), heading text, main text (BC). Usage: • Menu lists

	• Selection lists • Forms Similar to the previous layout, graphic added.
Setting item	Attribute text (ABC), value text (C). Usage: • Setting lists A two-line layout: the attribute text is on the top line, the value text (or graphic) on the bottom line in a box. The value box may be omitted in order to create a regular selection item within a setting list.
Two-line setting item	Main text (ABC). Usage: • Setting lists Used for setting-list items with no value text; typically a submenu item leading to another list of settings.



Figure 42: Single-line item



Figure 43: Single-line item with heading



Figure 44: Single-line item with graphic



Figure 45: Single-line item with graphic and heading



Figure 46: Single-line item with large graphic

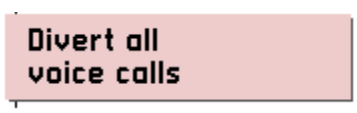


Figure 47: Two-line item



Figure 48: Double item



Figure 49: Double item with graphic





Figure 50: Double item with large graphic



Figure 51: Double item style 2



Figure 52: Double item style 2 with graphic



Figure 53: Setting item



Figure 54: Two-line setting item

6.3.3 Numbered items

Instead of small graphic, column A (where used separately) may contain a number. This can be used to indicate item numbers in lists where necessary.



Figure 55: Numbered list

Numbered items should be used only in lists where numbers are meaningful and give added value. There is no specific functionality (such as shortcuts) that all numbered lists would have.

6.3.4 Item type combinations

A list can only be composed of items of the same type. However, it is possible to use some item types for different purposes. An example is a setting list that contains an item for accessing a sub-list of settings. In this item, the value box is omitted so that the item looks like a regular selection item, and selecting it opens another list. Similar techniques are possible with other double item types, too.

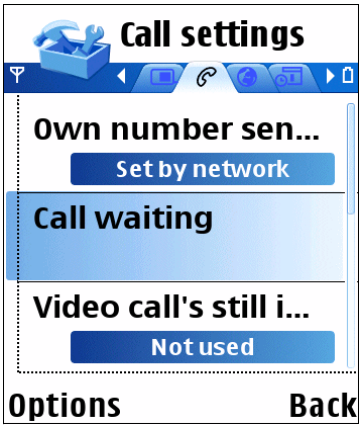


Figure 56: Setting list containing a non-setting item (Call waiting) to access another view

6.4 Grid Types

Grids are in many ways analogous to lists. However, there are some things worth noticing:

- In grids, the Arrow left and Arrow right keys are always used for moving the focus; they cannot be used in any other way that may be possible with lists.
- Grid layouts are not as standardized as lists; the layouts must be designed case by case for the applications. Typically, grid items occupy less screen space than list items. This results in grid items having fewer elements than

list items. A grid item may in general have one text, or one graphic, or a text and a graphic.

The following grid types can be used, and they are analogous to the corresponding list types:

Menu grid	For selecting one item; no **Options** menu.
Selection grid	Permanent state; can be left pending, the **Options** menu is available.
Markable grid	A selection grid with the marking function.

There are no grid types corresponding to a multi-selection list, a setting list or a form.

6.5 Find Pane

The Find pane is a component intended to help find items in a list. The Find pane is situated in the bottom part of the main pane.

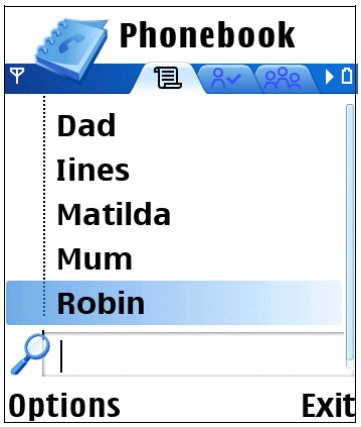


Figure 57: Find pane within a selection list of names in Phonebook

The standard functionality of the Find pane is as follows:

- Characters typed from the numeric keypad appear in the end of the string in the Find pane.
- The user can only add and remove characters in the end.
- The Find pane may be hidden until the user types in a character.
- Whenever the find string changes, the list in the main pane is filtered, and only the items matching the string are displayed. The user can browse the list normally using the Arrow up and Arrow down keys.

The Find pane can also be a pop-up window. In this case, it is only displayed when it is specifically activated.

6.6 Options Menu

The **Options** menu is a tool that offers the user a set of possible functions in the current context. Pressing the left softkey labeled **Options** opens the **Options** menu.



Figure 58: Options menu

The **Options** menu is a menu list displayed in a pop-up window. An item is selected by pressing either the left softkey (**Select**) or the Selection key. The user must either select an item from the list or cancel the menu; it cannot be left pending during another action (see Section 6.2.1).

The pop-up window is located above the control pane, and its height is dynamic; the maximum size is approximately the size of the standard main pane. The content on the screen outside the menu pop-up is dimmed.

Items in the **Options** menu use the single item layout; that is, they are text-only. The number of items in a menu is not limited; the list scrolls as necessary. The **Options** menu always loops; it gives the possibility to easily reach the end of the list.

6.6.1 Submenus

An item in the **Options** menu can be a submenu title, leading to additional choices that are displayed in another pop-up window (on top of the **Options** menu pop-up window) as a submenu.

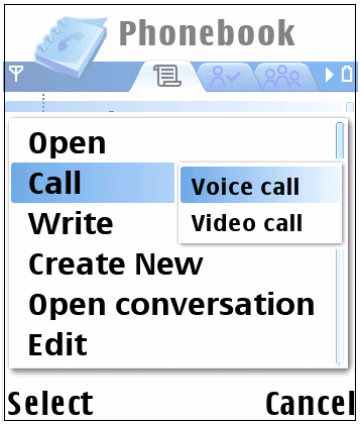


Figure 59: Submenu in the Options menu

The submenu is opened by pressing either the left softkey (**Select**), the Selection key or the Arrow right key.

The user can close the submenu window by pressing the Arrow left key. The main menu window remains open, with the focus on the submenu title.

When an item in a submenu is selected, both the submenu and main menu windows are closed.

The following rules apply to submenus:

- The number of items should be low, so that the user does not need to scroll in order to see all of them.
- Functions should not occur sometimes in the main level and at other times in a submenu. Items that are in a submenu should always be found in the same submenu.
- Only one submenu level is allowed, that is, a submenu cannot contain another submenu.

6.6.2 Submenu indicators

A submenu is at times used for changing the mode or temporary settings in applications. The submenu indicators are supported from S60 3rd Edition onwards. Radio buttons are used when only one item may be selected and tick marks are used where the user can choose more than one item. The indicators show the currently active mode or settings when the submenu is opened.

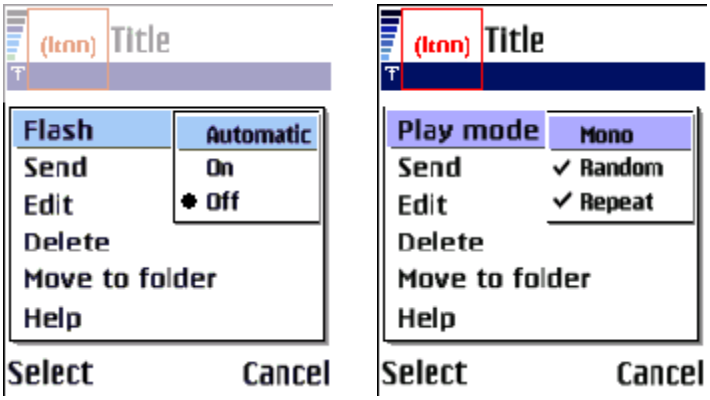


Figure 60: Submenu indicators: radio buttons (left) and tick marks (right)

6.6.3 Unavailable items

Situations often occur where a certain function cannot be used. In these cases, the corresponding items in the **Options** menu must either be hidden, or there must be an error message given when the user tries to access a function that cannot be accomplished. The S60 user interface does not use dimming of menu items.

This is a trade-off issue: removing unnecessary options makes the **Options** menus shorter, but at the same time it changes the menu from situation to situation, preventing users from learning the function locations. It may even cause frustration if the users expect some function to be found in the menu, but it is sometimes not there.

- In case the user has no reason to search for a certain function in the given situation, it should be hidden. As an extreme example, the **Delete** option is not needed when there are no items to be deleted.
- If the user searches for a function, even though it cannot be used in the current situation, it is often better to display the option and give an appropriate message if the user tries to access that function.

- An example of this is Digital Rights Management (DRM) protected files: for example, sending commands shall be visible in the user interface although sending such a file would not be allowed. Appropriate DRM specific notifications are displayed if the user attempts to select one of the options that the user is not allowed to perform on DRM protected media objects. This is to facilitate end-user's learning of the DRM concept.

6.6.4 Other Options menus

Certain **Options** menus are accessed in other ways than by using the **Options** softkey. The usage of these menus is similar to the normal **Options** menu. Examples of such menus are:

Context specific **Options** menu	The Selection key opens this context specific **Options** menu when there is no single intuitive function (such as opening the item in focus) for it. The context specific **Options** menu only lists functions that: • Affect only the item in focus. • Could be regarded as potentially intuitive. • Are competing for the topmost place of the menu. Other functions can be accessed through the **Options** softkey. The number of items in the context specific **Options** menu is usually two or three; it should never be more than four. As an example, in a message editor, the functions in the context specific **Options** menu could be **Send** and **Add recipient**. In case there are marked items in a list, the context specific **Options** menu should include the **Mark/Unmark** functions. When a list is empty, it may make sense to offer a **Create new** type of option in the context specific **Options** menu, but only when it would be an appropriate function in the context.
Editing menu	Opened by pressing the Edit key in a text editor. Contains only editing commands. See Section 8.6.

6.6.5 Options template

The order of items in an **Options** menu should follow the template presented below (in specific cases, when there are strong arguments against the order in the template, the order can be changed.)

The option names listed here are generic names, not the actual texts used in the products. The texts may even vary between applications even though the logical item is the same.

Items that should appear in every full **Options** menu opened from the left softkey are labeled mandatory. However, these items are not required in context specific **Options** menus.

For other than the mandatory items, only the items needed in each context shall appear in the **Options** menu.

Items specific to the context can be added among the common items, in the places where they best fit, considering the importance and probable usage frequency. The places where context-specific items may appear are represented as **+++** in the list.

Submenu titles are indicated with ▸ and are followed by the submenu items.

Open / Select /Change		Mandatory when the Selection key performs an **Open**/**Select** function. **Change** is used in setting lists. Same as the Selection key function.
+ + +		
Call now		For immediate calling when a phone number is available (typically highlighted). For example in Phonebook, the phone number is in focus.
Send now		• Immediate sending in a message editor, when an address exists. • Opens a message editor when there is an address available.
Send		Open a list query that contains the following Send options.
	SMS	Start writing a new short message.
	MMS	Start writing a new multimedia message.
	email	Start writing a new email.
	MMS Upload Services	Service provider item for uploading.
	Bluetooth	Send objects via Bluetooth.
	IR	Send object via Infrared.
	Other installed MTMs	Send objects via some other available MTM.
Create new		Initiate creation of a new item. When more than one type of item can be created, a submenu may be used to select the type.
+ + +		
Create new		For initiating the creation of a new item, e.g. a message in the Messaging Centre application or a calendar event in Calendar. When more than one types of items can be created, a submenu should be used for selecting the type.

Edit item		Enables editing of the current item, for example a form, or an individual item on a list.
Delete item		• Deletes the item(s) in focus (or marked) on a list. • Deletes the current item being viewed.
+ + +		
View info (1)		View detailed information about the current item. In this location in applications where this is a high-priority function.
+ + +		
Move		Move an item to a different location within the list or grid.
Move to folder		Move item(s) into a folder.
New folder		Create a new folder.
+ + +		
Edit list ▸		A submenu for functions used in markable lists.
	Mark / Unmark	Mark or unmark the current item, depending on the current state.
	Mark all	
	Unmark all	
Rename		Rename the item in focus.
+ + +		
Add to contacts ▸		A submenu for functions used to add contact information into Phonebook.
	Create new	Create a new contact item.
	Update existing	Add new field(s) to an existing contact item.
Find item ▸		A submenu for functions used to extract contact information from text in viewers and browsers.
	Phone number	
	Email address	
	URL	
+ + +		
View info (2)		View detailed information about the current item. In this location in applications where this is a low-priority function.
Add to Pinboard		Add a link to the current application or document into Pinboard.

Settings	Access to application settings or context-dependent settings.
Help	Open Context Sensitive Help.
Exit	Mandatory. Exit the application.



Note: Prior to S60 3rd Edition, pressing the **Send** item opened a submenu instead of a list query.

6.7 Notes

A note is a feedback component that informs the user about the current situation. A note contains a text and possibly a graphical element. The softkey labels are typically empty (a Wait note is an exception to this).



Figure 61: Information note

Notes do not require user input, although a user can dismiss most notes by pressing any key.

The following note types are in use:

Confirmation note	Informs the user about a successfully completed operation. Short duration, a subtle tone (this should not be used after every kind of successful action; see the guidelines below).
Information note	Gives information about an unexpected situation during the usage of the device. Longer duration and a more noticeable tone than in a Confirmation note. Errors that are not too serious should cause an Information note.
Warning note	Used when the user must be notified about something that may require action. Fairly long duration, and a sound that can be heard (and distinguished) even when not concentrating on the phone. For example, the battery low warning.
Error note	This is a warning to the user. It should only be used when the user has tried to do something that may cause a considerable problem. See the guidelines below.
Permanent note	A note that must remain on the screen for an indefinite time. The user cannot dismiss it. For example, **Insert SIM card**.
Wait note	A note containing a progress or wait graphic (a wait graphic is an animation of indefinite duration, whereas a progress graphic is a growing bar that can be used when it is possible to estimate the duration of the operation). Used during operations that take a long time. Whenever possible, the user should be able to stop the operation. For this, a softkey labeled e.g. **Cancel** is provided.

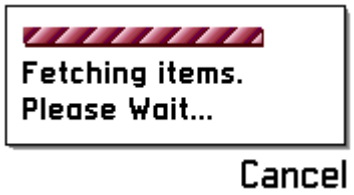


Figure 62: Wait note with the Cancel function in the right softkey

Here are some guidelines concerning note usage:

Use a Confirmation note when:

- The effect of the operation cannot be seen directly by some other means. For example, Message sent.
- There is some relevant information to be communicated by it. For example, Last call duration.

Confirmation notes should not be used after every completed operation, as this would easily start to annoy users. Confirmation notes should not be used when:

- There is already another dialogue in the procedure, for example **Do you want to remove this message? Y/N**.
- A progress indication is visible during the procedure.
- The user can see the result of the operation when it is performed. For example, when adding or removing objects in a list.
- A setting has been changed. The new value of the setting is visible in the setting item.
- The operation can be considered minor or so frequent that a note would be annoying. For example, Copy-Paste actions.

Use an Error note when:

- The user does something that may cause considerable harm immediately or later. Example: the user gives a wrong PIN code. Repeating this a couple of times would block the SIM card.

To keep Error notes effective, they should be used very sparingly. In most ordinary error cases, an Information note should be used instead of an Error note. It has a less aggressive sound and graphics.

It should also be noted that if the information to be given is such that the user must see and acknowledge it, a Confirmation query is a better component to use than a note. Then the user must press a key to dismiss the information, and there is time to read and think about the notification.

6.8 Soft Notifications

Soft notifications are reminders that inform the user of events that have occurred in the user's absence, or while the user was busy with some application. Text, and also graphics, can be used to communicate the message to the user. Soft notifications are displayed in pop-up windows.

The user can respond to the soft notification by using the softkeys. The left softkey is used for activating a function, for example opening a message that has arrived. The right softkey is used to discard the notification without taking any further action.



Figure 63: Soft notification indicating the arrival of voice messages

Soft notifications are displayed only in the Idle state. If an event that causes a soft notification (for example a missed call) occurs when an application is active, it may cause other kinds of UI events to notify the user, but if the user does not react to these, the soft notification appears only after the phone is put in the Idle state if the event still requires it.

The application that launched a soft notification can control it and also discard it. It is possible to use the Applications key during a soft notification; in that case, the soft notification disappears, but reappears when the user returns to the Idle state, unless the application responsible for the notification has discarded it.

Soft notifications can be displayed for the user in two different appearances:

- **Ungrouped soft notification**: These notifications contain one piece of information each. The appearance of the notification window is the same as a note's. The example in Figure 63 is an ungrouped soft notification.
- **Grouped soft notification**: Many different items of information can be combined into one soft notification where the items are displayed as a list. The user can pick up one of the items at a time and react to it. The appearance of this soft notification type is the same as the appearance of a List query (see Section 6.9).



Figure 64: Grouped soft notification. The height of the window is dynamic and depends on the number of lines in the list.

6.8.1 Discarding soft notifications

The application that launched a soft notification can discard it without user intervention when the notification becomes obsolete. A soft notification should remain pending until the user has responded to it, or started using the corresponding application so that in effect the notification becomes obsolete. In that case, the application can discard the notification even though the user may not have actually seen it.

When the user reacts to a soft notification by pressing the left softkey (for example **Read**), or selects one item of a grouped soft notification, the item becomes interpreted as obsolete, and will not reappear. If a soft notification contained more than one item, the other ones remain pending and reappear when the user returns to the Idle state.

The user can dismiss the notification by pressing the right softkey, labeled **Exit**. After this, the notification does not reappear until new events cause a new notification to be created. In case of a grouped soft notification, all items it contains are discarded.

Please note that discarding a soft notification does not mean that the received content is lost.

6.8.2 Many simultaneous soft notifications

Soft notifications are stacked in case there are more than one pending at a time: after the topmost one is discarded, the one following it will be displayed. Each notification has a priority value that determines the order of the notifications.

6.8.3 Information pop-up

Information pop-up notes are used to provide additional information to the user, most typically regarding a focused item. Although the information pop-up can be used in various applications and components, it should not be used too frequently as constantly appearing and disappearing notes may disturb smooth user experience.

The information pop-up does not have focus, its content cannot be scrolled and the control always remains in the main pane. The maximum number of rows in the information pop-up is four. In a typical case, the pop-up emerges after one second the user has taken the focus on an item but has not made any further actions. The pop-up is timed so that it (typically) remains visible for ten seconds or until some event interrupts it. Timeouts are application-specific and variation is possible.

6.9 Queries

A query is a state where the software waits for user input. Queries are used in situations with more than one ways to proceed, when the application needs data from the user, or when it is necessary to make sure the user knows what is happening. A query must be exited before the application can proceed.

Queries are displayed in pop-up windows. The following query types can be used:

- Confirmation query: a question or notice with one or two possible responses.
- List query: a question with a selection of more than two possible (predefined) responses.

- Multi-selection List query: presents a list of items; the user can select any number of them.
- Data query: used for numeric or textual input.
- Message query: used for various purposes in queries that are longer than one screen

6.9.1 Confirmation query

A confirmation query requests the user to confirm an operation, or asks a Yes-No-type of question. It can be used e.g. to make sure that the user does not accidentally delete important information or start an operation which cannot be canceled.

The layout for a Confirmation query is the same as the note layout, with an optional graphic item.

One or two softkey labels can be used. In case the query can cause two different consequences, the positive choice (**Yes**) is placed on the left softkey, and the negative one (**No**) on the right softkey. In pure confirmations, only one way to proceed is possible, and the response text (for example **OK**) is placed on the left softkey.

The Selection key always causes the same action as the left softkey.



Figure 65: Confirmation query

Guidelines for designing Confirmation queries:

- When designing the prompt text, make sure that the positive answer is also a safe one. Users tend to proceed pressing the left softkey or the Selection key without thinking too much.
- Redundant Confirmation queries should be avoided. Do not add a Confirmation query if there already are other forms of feedback, unless it is crucial that the user gets a certain piece of information.

6.9.2 List query

A List query offers a list of predefined choices for the user. It can be used when more than two options must be offered to the user. There is a prompt text (optional) on top of the window, and a list of options to choose from.

The list in a List query is a menu list: the user can select an item or dismiss the query; the **Options** menu is not available. The default softkey labels are **OK** on the left and **Cancel** on the right, the actual texts can be specific to the context. The Selection key causes the same action as the left softkey.

Any list item layout suitable for menu lists can be used; see Section 6.3.



Figure 66: List query

The number of items in the list should be kept low, so that all items can be seen without scrolling.

Instead of a list, a grid can be used in a query. The grid query function is otherwise identical to a List query.



Figure 67: Grid in a query

6.9.3 Multi-selection List query

A multi-selection List query is used when the user needs to be able to select several items from a list at the same time. See Section 6.2.4 for a description of a multi-selection list.

The left softkey (**OK**) is used for accepting the query, and the right softkey is **Cancel**.

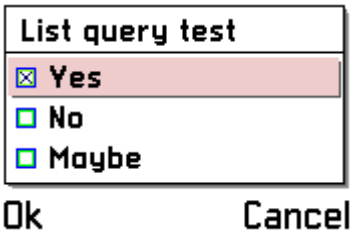


Figure 68: Multi-selection List query

6.9.4 Data query

A Data query requests the user to type in some alphanumeric or numeric information, such as a name or a phone number.

The query contains a prompt text and a user input field. The input field can have any type of an editor, depending on the context, so that the input may be restricted to e.g. numeric data, date or time only. Both the prompt and input fields can be longer than one line when necessary.

The softkeys of a Data query are **OK** on the left for accepting the input, and **Cancel** on the right softkey for discarding the query. The Selection key accepts the input in the same way as the left softkey.

The Clear key is used for deleting characters and for that purpose only.

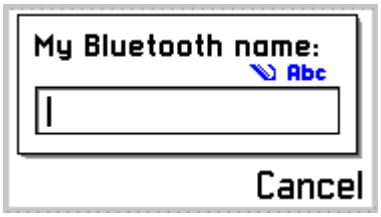


Figure 69: Data query. The input field may extend automatically to more than one line's height, and also the prompt text may take two lines.

6.9.4.1 Password query

A specific case of the Data query is the Password query, used for confidential information such as passwords or PINs. It uses the password editor, which functions much like any other editor, except that instead of the actual data, a dummy character (asterisk) represents each input character. In case of a numeric-only password, the asterisks appear as soon as characters are entered. In case of alphanumeric input, to support typing characters by repeated presses of the same numeric key, the character is displayed normally for a short time, and then changed into an asterisk.

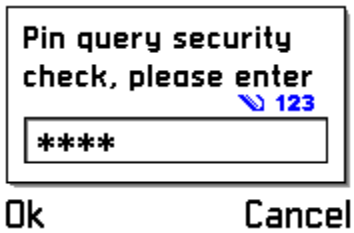


Figure 70: Password query window

6.9.4.2 Data queries with multiple fields

It is possible to have two input fields in a Data query. An example of this is a user name and password query: one field is a normal alphanumeric editor and the other a password editor. In this case, a press of the Selection key moves the insertion point from the first field to the next; in the second field it accepts the query. The left softkey always accepts the query. The user can also move from one field to the other using the Arrow up and Arrow down keys.

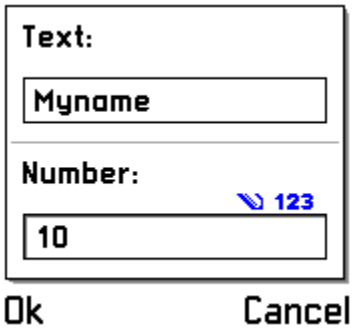


Figure 71: Data query window with two input fields

6.9.5 Message query

When there is need for a query that is longer than would fit one screen, a message query can be used. A message query can contain a link text that can be highlighted and opened.

By default the left softkey is **OK**, but with a link in the query the left softkey opens the target view of the highlighted link text. The Selection key also opens the link target view when the link is highlighted.



Figure 72: Message query with a link

6.10 Indicators

Indicators are graphical or textual objects on the screen that provide information about the status of the system. They cannot be used for input, and there is no focus on an indicator: the user cannot point at an indicator to perform actions.

The various indicator types in use are described in the following subsections.

6.10.1 Signal and battery indicators

The top-left area of the screen is used for the cellular signal strength indicator, and the corresponding top-right area for the battery level indicator.



Figure 73: Signal and battery indicators on the sides of the status pane. Other status indicators are visible below the navi pane.

The signal indicator is a part of the status pane, and it is displayed in all states where the status pane exists. The indicator consists of an antenna icon and a bar graph that indicates the current signal level. The antenna icon may be replaced by another icon indicating the GPRS connection status.

Note: From S60 3rd Edition onwards, the battery indicator is displayed in the status pane also in the application menu and within applications. In earlier S60 platform versions, the battery indicator was only displayed in the Idle state in the Phone application.

6.10.2 Status indicators

Status indicators are small graphical icons. They inform the user about such issues as unread messages, waiting voice mail, selected phone line, IR and Bluetooth connection status, set clock alarm, home zone, and locked keypad. Each status indicator has a priority number, which determines which icons are displayed in case there are more of them than fit on the screen simultaneously.

In the Idle state and in the Phone application, the status indicators are displayed in the status indicator area below the status pane (see Figure 73).

Within applications other than Phone, status indicators appear in the universal status indicator pane, which is the same area as the battery pane. Due to the small size of this area, and also to avoid displaying unimportant things in general, only the most important status indicators are displayed there. This is true even in cases where an indicator could fit into the pane: only the most important ones appear as universal status indicators, others are only seen in the Idle state.



Figure 74: Universal status indicators displayed in the top-right corner

6.10.3 Indicators in the navi pane

The navi pane can contain several kinds of indicators, depending on which way the pane is used in the particular context:

Left and right arrows	When horizontal navigation between different main pane views is used, the navi pane displays the navigation information (tabs or text may be used). The arrow icons on the left and right ends of the navi pane indicate the possibility to move in the corresponding directions (with tabs, the arrows are only displayed when all tabs are not visible).
Editing indicators	When an editor is in use in the main pane, the indicators related to editing parameters are displayed in the navi pane. They indicate things such as the editing mode (numeric/alphanumeric), character case, Predictive Text status, and the available space.
Audio volume indicator	Used in applications where audio volume adjustment is necessary such as Phone.



Figure 75: Arrows in the navi pane



Figure 76: Editing indicators in the navi pane



Figure 77: Volume indicator in the navi pane

6.10.4 Editing indicators in pop-up windows

When a pop-up window contains an editor that needs editing indicators, it is not feasible to use the navi pane for the indicators. For this purpose, a specific area in the pop-up window, above the editing field, can be used for displaying the editing indicators (see Figure 39 and Section 8.6 for an example).

6.10.5 Soft indicators

Soft indicators are textual indicators displayed in the main pane of the Idle state. Examples of cases where a soft indicator can be used are call charges indication and MCN (Micro-Cellular Network) area indication.



Figure 78: Soft indicators in the main pane

6.10.6 Operator indicator

In the Idle state, the title pane contains the operator indicator. It is either a text or a graphical image.

6.10.7 Application-specific indicators

Applications may need indicators specific to their own purposes. Such indicators can be placed into the navi pane, if it is available, or into some part of the main pane. However, using the main pane this way may require the use of a specific main pane layout. Notice also that the icons present in many list item layouts can be utilized as indicators.

7. S60 UI with Side Softkeys

S60 UI can rotate between the portrait and landscape layout (supported from S60 2nd Edition, Feature Pack 3 onwards). This chapter describes the guidelines for designing landscape-aware application user interfaces. On the whole, the primary guideline in S60 platform is that no specific UI design is necessary for landscape orientation and mode. Instead, applications designed with the guidelines described in this chapter should work in both landscape orientation with side softkeys and with the default portrait orientation. This chapter does, however, describe some exceptions to this rule.

	**Note:** Built-in S60 platform applications support both portrait and landscape from S60 3rd Edition onwards. In S60 2nd Edition, Feature Pack 3 only applications using only AVKON components run in both portrait and landscape orientations automatically.

7.1 Introduction to the Side Softkey Style

7.1.1 What is S60 UI with side softkeys

S60 UI with side softkeys assumes a hardware that has the softkeys on either side of the display. A product concept for such a case might be, for example, a device optimized for using some specific applications in a 90 (or 270) degrees rotated mode, but not having extra keys for providing the softkey functionality in the bottom of the display in this rotated mode. Note that the landscape orientation also works with bottom softkeys in devices with that type of an aspect ratio.

Thus, the softkeys in this mode are on either side of the UI depending on the hardware and its mode.

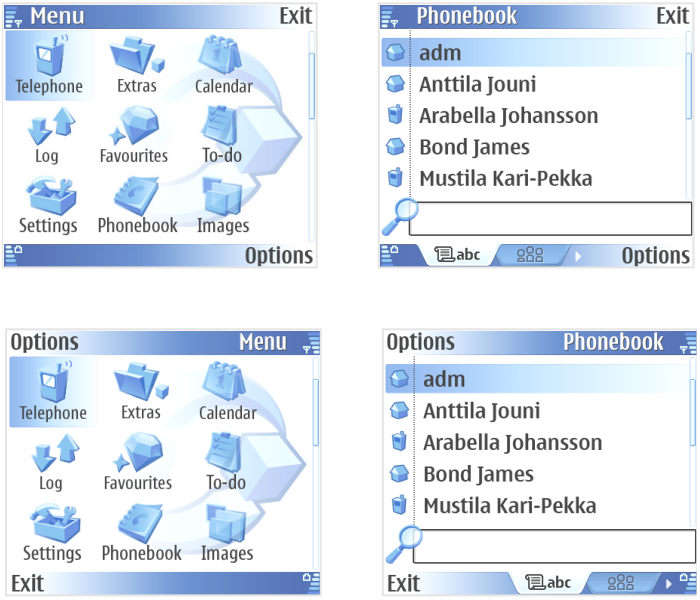
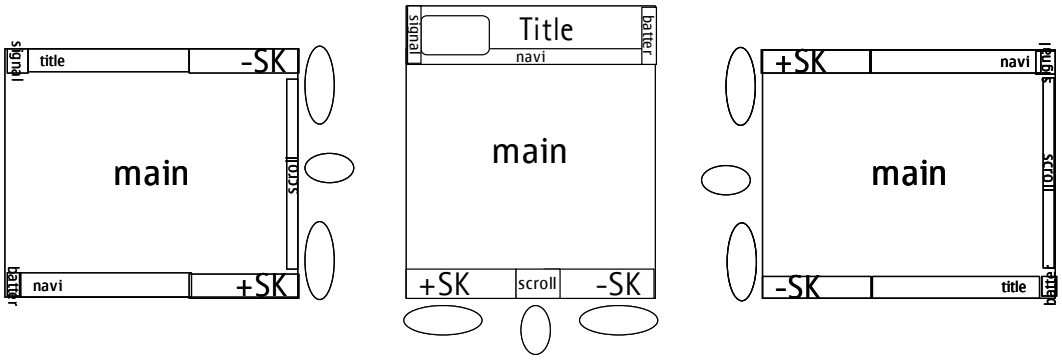


Figure 79: Example images of S60 UI with side softkeys

The side softkey style assumes the same keys as the default S60 UI.

These style guidelines make no assumptions about the display resolution, about the exact aspect ratio between the height and width of the UI, or about the pixel density of the UI. However, the assumption is that the side softkey style is applied in either a landscape (width > height) or a square (width = height) display mode, but not in a portrait display mode (width < height).

7.1.2 Default layout description and the basic principles of rotation



Series 60 UI with side softkeys on the right

Default series 60 UI with softkeys below the display

Series 60 UI with side softkeys on the left

Figure 80: S60 UI left and right side softkeys with legacy layout. Note: In a rotating UI, there is a scroll bar even in the legacy S60 UI.

The style for S60 UI in the landscape mode with the side softkeys is based on the following core principles:

- The height of the main pane is maximized with the cost of reducing the sizes of the other panes, with the principle of optimizing the content rather than the status information.
- The status pane and the control pane are not handled as entities, but they are divided into their sub-panes that may be positioned into different locations of the UI. The present style guidelines assume that the panes are located as presented in Figure 79. However, S60 platform supports flexibility for the actual product implementation and, therefore, the UI design shall not assume any certain position for the panes. There is no support for a context pane icon in the side softkey style.
- In the side softkey style, the control pane scrolling indicators are always replaced with a scroll bar. In a left-to-right display language the scroll bar is always located on the right-hand side of the main pane, regardless of where the softkeys are.
- The relation between the physical softkeys and their labels in the SW UI remain in the 90 and 270 degrees rotation: the same key that was **Options** before rotation is **Options** also after the rotation, and so on (see Figure 81). **Note:** This behavior may be product concept dependant, but it must be consistent within one device mode.

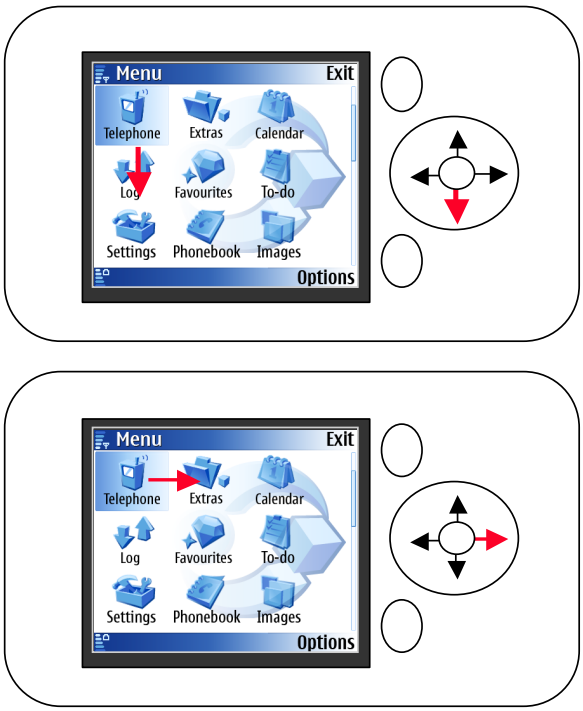


Figure 81: Relation between the physical softkeys and their labels in the SW UI in 90-degree rotation

- The interaction design for any S60 application should be as flexible as possible. A generic application should no assume anything about the position of the softkeys or panes; that is, not assuming a certain position for softkeys, not expecting direct relation or proximity between the title and navi panes, and so forth.

7.2 Interaction

7.2.1 Rotation

By default, the rotation from the legacy style to the side softkey style is global. Whenever the orientation of the UI is changed, all the applications rotate 90/270 degrees (not only the foreground application). The user interface does not by default rotate automatically to any other orientation in cases of opening or closing an application, swapping to another application, or in a case of embedded applications or seamless task flows. In addition, the UI should not change its orientation when another view within an application is opened (see Figure 82).

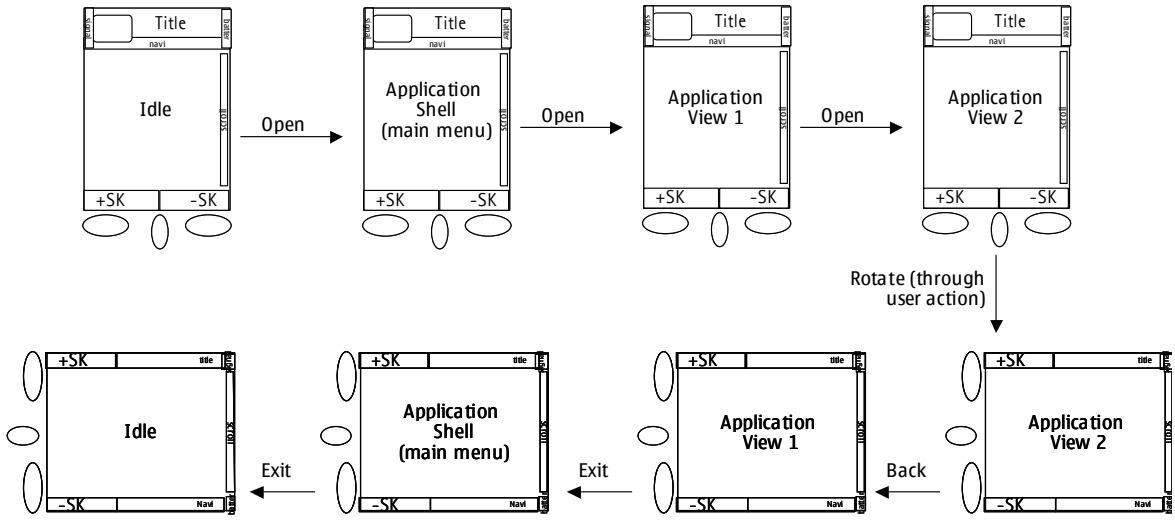


Figure 82: Orientation remains until the user changes it, i.e. the UI does not by default rotate automatically to any other orientation in cases of opening or closing an application, etc.

There may also be applications by S60 platform licensees or by third parties that support only one orientation. For such applications, there should be an indication in Application Shell: an icon to indicate that the application opens in a different orientation than the current. If such an application cannot be launched at all for the current rotation (e.g. if the rotation requires a physical switching of the product), there must be an indication for that as well.

Applications that support only one orientation, and thus trigger automatic orientation switch, do not change the orientation of other applications. That is, if the orientation is 'portrait' before launching a 'landscape-only application', it is portrait also after exiting it (see Figure 83).

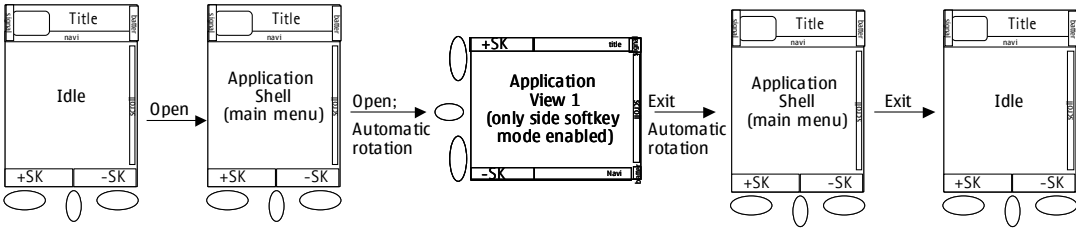


Figure 83: Orientation changes in a case of an application that supports only one orientation

An exception to this rule is applications that are launched from the landscape application itself (e.g. the **Go to gallery** option in the Camera application) or embedded within the application (e.g. the embedded Browser within an image editor when downloading content). In these cases, one orientation should remain throughout the task flow, and the original orientation should be returned only after exiting the application that triggered the rotation (see Figure 84).

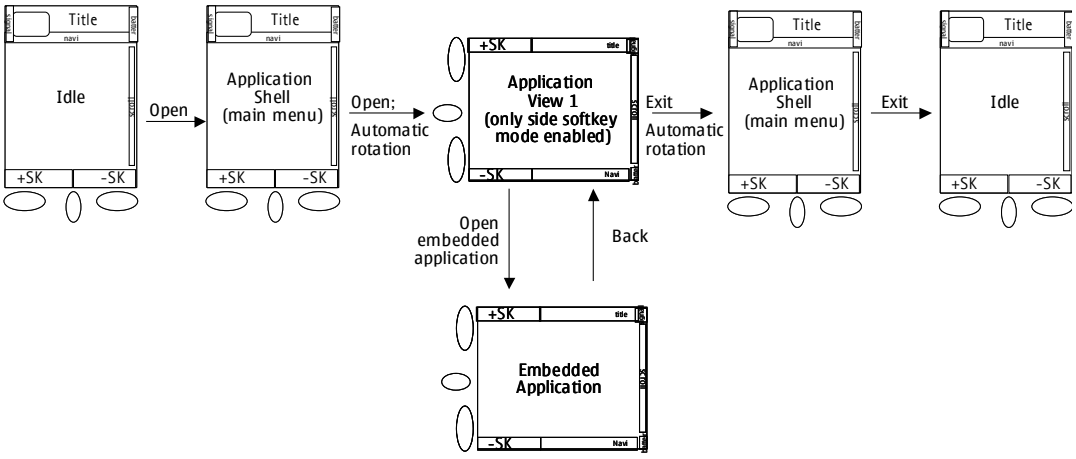


Figure 84: In case of an embedded application view within a one orientation only application, the orientation remains throughout the task flow, and the original orientation should be returned only after exiting the application that triggered the rotation.

No queries or notes are recommended to inform or ask the user about rotation.

7.2.2 Basic interaction style

The basic interaction principles of S60 platform in the side softkey style remain similar to the legacy S60 UI style. This applies to keypad functions, typical functions of the keys, navigation (both in terms of hierarchies and the scroll and select behavior), etc.

As in the legacy UI style, the scrolling in the software follows the visual direction of the respective scroll keys.

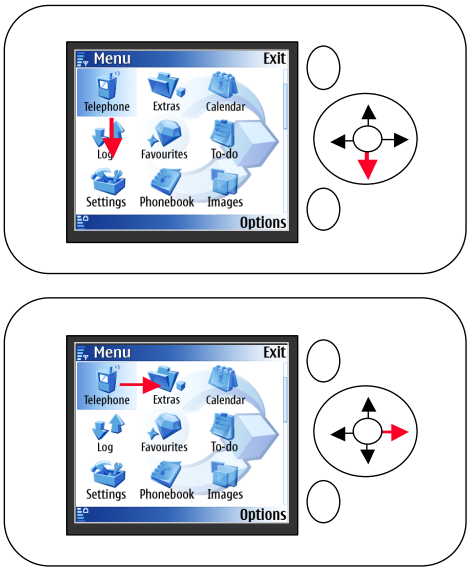


Figure 85: Pressing a scroll key moves the focus in the expected direction

Similarly, other visual UI controls in scroll keys rotate together with the UI. For example, this applies to volume control in media applications and zooming in visual media applications: The key that is pointing upward after the rotation zooms in, and the key pointing currently left reduces the volume level if such an

application is on the foreground, following the visual UI components (see Figure 86).

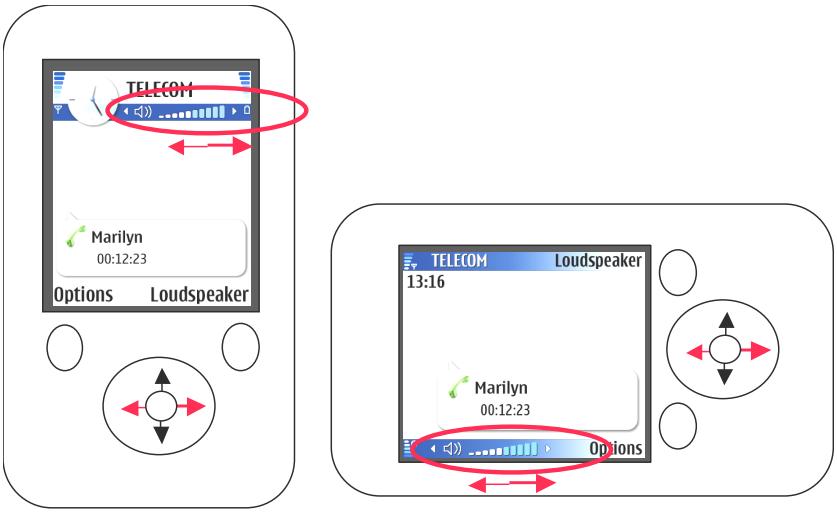


Figure 86: Visual UI controls in scroll keys rotate together with the UI

	Note: In some product concepts, using the numeric keys and some of the navigational keys may be disabled while in the side softkey mode. This may mean that some applications or features (such as editing) are not usable at all in one of the modes.

7.3 Designing Application UI for the Layout

7.3.1 Main pane

The height of the main pane is maximized with the cost of reducing the sizes of the other panes, with the principle of optimizing the content rather than the status information. Therefore, the other panes are tailored to give more room for the main pane height.

The full width of the display (the longer side) is used for showing the main pane content. On the right-hand side of the display (in left-to-right display languages), there is a scroll bar component for indicating the scrolling.

Lists	In terms of display height, the amount of lines or items may change in the rotation. Thus, a generic UI design does not assume a certain fixed number of items. In terms of width, there is more space for content in the rotated mode assuming a landscape display. In lists, this shows mainly in localization. There may be a separate localized string for landscape views with wider localization metrics. The virtual column structure of the main pane lists remains. By basic principle, the additional width is added to the C column, i.e. to the main text content column.
Grids	In terms of display height, the amount of lines or items may change in the rotation. Thus, a generic UI design does not

	assume a certain fixed number of items. In terms of width, there is more space for content in the rotated mode assuming a landscape display. Designing a grid view for landscape is decided individually for each application.
Pop-up windows	The overall size of pop-up windows remains the same in both orientations. The placing of the pop-up windows is adjusted in the landscape mode so that they remain in close proximity to the softkey texts. **Note:** Due to the rotation and its impact on the main pane height, it is important that the rules regarding the amount of items in context sensitive **Options** menus and in submenus are respected, although in certain portrait variants there may be more space available in terms of height.

There are no changes to the way highlight, scrolling, grid or list filling, or other normal UI operations behave.

7.3.2 Status pane and the context pane

In the side softkey style, the status pane is not handled as one entity, but through its sub-panes: the title, navi, signal, battery, and universal indicator panes.

There is no support for a context pane icon in the side softkey style. Thus, when the UI of an application is rotated, the context pane icon is ignored: It is shown only in the legacy S60 style and layout.

Still, if the application is to be used in the legacy style (within the same product concept or across a range of products), the context pane should be designed.

If an application by its nature requires using the context pane for showing some essential data, an alternate rotated layout for such applications needs to be redesigned. Such a case may be, for example, Video Telephony (the context pane shows the caller's image), Phonebook (thumbnail image), or Idle state (clock).

7.3.3 Title pane

By default, the title pane is placed above the main pane.



Figure 87: Example of the title pane above the main pane

In the side softkey style, the title pane has only one line for the text. The title pane shares space with the upper softkey and universal indicators. The possible cases of overlapping are managed through transparency and font or background color adjustment. These effects may be UI theme dependant. If there is not enough space for showing both the softkey label and the title, the title is truncated by default.

Due to these space limitations, there are some requirements for designing titles for application views:

- There are no titles assuming two logical lines of text. For example, indications and other similar items in the title pane are moved either to the navi pane, main pane, or other panes.
- Titles that require a long space for the content (e.g. titles that in the default portrait layout tend to require two lines of text) should be avoided. This can be done by moving such content to the main pane (for example, showing a contact name as fields in the main pane instead of trying to fit it to the title) or to other panes. Such changes to existing applications should be done to the default design, so that no separate layout or component design is needed for the different rotated and non-rotated variations of the UI.

7.3.4 Navi pane

In the landscape mode, the navi pane is positioned below the main pane.



Figure 88: Example of the navi pane below the main pane

The navi pane shares the space with the lower softkey. There should be enough space reserved for the panes so that they do not overlap, and thus no specific logic is needed to manage the overlapping.

7.3.5 Signal pane

The signal pane is always in the upper status area next to the title pane and adjusted away from the softkey.

7.3.6 Battery pane

The battery pane is always in the lower status area next to the navi pane and adjusted away from the softkey.

7.3.7 Universal indicators

Universal indicators are placed dynamically in the upper status area between the title and the upper softkey panes. The possible overlapping cases are managed through transparency and font or background colors. These effects may be UI theme dependant.

The space allocation between the title, universal indicator, and upper softkey panes is arranged so that there is space to show a minimum of two of the highest priority indicator icons (one column) without overlapping. More indicator icons may be shown, but the title may overlay transparently.

7.3.8 Control pane vs. softkey panes

Similarly to the status pane, the control pane is not handled as one entity but through its sub-panes: the upper softkey pane and the lower softkey pane. Scrolling indicators are replaced with a scrollbar component that is placed on the right-hand side of the main pane.

The relation between the physical softkeys and their labels in the SW UI should remain in the 90 and 270 degrees rotation. The same key that was **Options** before rotation is **Options** also after the rotation, and so on (see Figure 80 and Figure 81) (**Note:** This behavior may be product concept dependant, but it must

be consistent within one device mode). Thus, in a mode with softkeys on the right, the upper softkey is positive and the lower softkey is negative, while in a mode with left side softkeys, the lower softkey is negative and the upper softkey is positive.

8. Applications and Design Examples

8.1 Idle State

The Idle state is the basic state of the device. It displays information about the current state of the device, including the connection status, battery level, and network operator. See Chapter 6 for detailed information about the indicators.



Figure 89: Idle state

The functions of keys in the Idle state, when Active Idle is not activate, are as follows:

- The user can configure the Selection key, the softkeys and the navigation keys to access various applications.
- The Numeric keys can be used to dial a phone number manually; a press of a Numeric key opens a number entry window. A long key press in the Idle state is used for one-key calling to the number assigned to the key.
- A short press of the Power key opens the list of Profiles for switching the active profile.
- The Applications key enters Application Shell.

The user can set any image as the background wallpaper in the Idle state.

8.1.1 Active Idle

The user can have additional functionality and see more information, such as Calendar data, in the Idle State by activating Active Idle (introduced in S60 2nd Edition, Feature Pack 2). When Active Idle is active, the Selection key and the navigation keys are used to navigate and control the Active Idle view and thus cannot be configured to access other applications.

Active Idle is placed on top of the background image in the basic Idle state. It comprises a focusable row of icons (which by default are shortcuts to applications), row(s) for Calendar events, To-Do items, player, SAT and NewsTicker (device-dependant). The application icons and the Calendar content (as many items as there is room for) are always visible. To-Do, player and SAT are visible when there is information to show, and NewsTicker is always visible if enabled from the settings.



Figure 90: Active Idle

8.2 Application Shell

Application Shell is the menu used for accessing all applications. Pressing the Applications key opens Application Shell, where the user may browse and select an application. Applications are presented as a grid of items by default; the user can also choose to use a list presentation. Pressing the Selection key opens an application and closes Application Shell. Refer to Section 8.3 for detailed information about opening and closing applications.



Figure 91: Application Shell basic 3x3 portrait grid



Figure 92: Application Shell with a 3x4 grid in portrait (left) and landscape (right)

As well as the basic 3x3 grid layout, the Application Shell icons can be placed on a 3x4 grid (from S60 3rd Edition onwards), in which case the title pane on the top of the screen is adjusted (see Figure 92) to allow for more space. For more details on the landscape mode, see Chapter 7.

8.2.1 Shortcuts

When Application Shell has been opened, the Numeric keys 1-9 can be used as shortcuts to selecting an application. The keys are mapped directly to the nine icons in Application Shell's initial view, so that the Numeric key 1 corresponds to the top-left application and the Numeric key 9 to the bottom-right one.

8.2.2 Customizing Application Shell

The user can adjust the order of applications, as well as create folders and move applications into folders within Application Shell. These managing functions are available through the **Options** softkey on the left.

8.2.3 Fast Application Swapping

There is a shortcut to swapping between applications that are currently running. A long press of the Applications key opens the Fast Application Swapping pop-up window that contains the icons of the currently running applications. The user can browse and select one of these in the same way as in Application Shell.



Figure 93: Fast Application Swapping window

8.3 Application Handling

This section describes the handling of applications and the interactions between applications in the S60 user interface environment.

Some basic assumptions:

- There can be only one instance of each application at a time. This means that when the user selects an application in Application Shell, there is never a confusion about which process it refers to: either there is one running instance of the application, in which case it will be displayed, or there is none, in which case a new process will be created.
- However, software modules that several applications can use (such as editors) may run simultaneously in more than one application. The user may thus see the same feature being run in several different applications at the same time.

8.3.1 Opening and closing applications

Applications are typically opened using Application Shell, the menu containing all the installed applications.

There may be other ways to open an application, such as:

- Using a shortcut in the Idle state.
- Using a link in Pinboard.
- Using a specific shortcut built into another application.

When there is no instance of the opened application already running, a new process for the application is created. If the application is already running, opening the application means bringing the existing application process on top. In case of a link that points to a specific state in an application, the existing application is interrupted and forced into the target state. See Section 8.3.2 for more information on this.

The user can close applications in the following ways:

- Using the right softkey, which goes backwards in the application hierarchy (**Back**) and finally exits the application (**Exit**).
- Using the **Exit** function from the **Options** menu.
- Pressing the Call termination key (from S60 2nd Edition Feature Pack, 3 onwards).

Closing an application means that the processes associated with it in the working memory are terminated.

Application processes can also be terminated by the system, for example when the user powers down the device.

8.3.2 Multitasking

The S60 UI style allows multitasking; that is, working with more than one application simultaneously. To accomplish this, an application can be left running when switching to another application, and it is possible to swap between running applications and interact with them.

To open another application without terminating the current one, the user can press the Applications key to go directly to Application Shell, and select the other application from there. The first application process then remains running in the background while the user is interacting with the second application.

It is possible to swap between two or more running applications by using Application Shell.

It is also possible to go to the Idle state and leave an application running by selecting the Idle state from the Fast Application Swapping window.

The number of simultaneously running (different) applications is limited only by the available memory in the device. When a new application process cannot be created because of limited memory, the system can automatically shut down applications to gain more memory space.

8.3.3 Application interactions

There are two different models of interaction between applications:

1. Use of modules (services) that can be called and run within several different applications. In this model, a service or library function is running within the application the user was originally working with. The applications do not conflict with each other when this model is used. It should be noted that from the user's point of view, each item in Application Shell is seen as an application. A specific service can be run in any number of these applications at the same time, so the user may see a similar screen in many applications running simultaneously. However, the user cannot launch a new instance of any of the applications from Application Shell before terminating the existing one first: selecting a running application will simply revert to the existing one. The **Back** function works normally in this model: the user can step back from an embedded module to the calling application; the modules may even be nested.

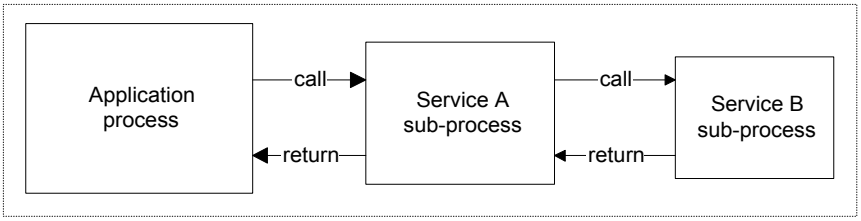


Figure 94: Nested services running within an application process. The Back function leads to the parent process.

2. Actually switching from one application to another to accomplish a task. In this model, the other application may need to be interrupted if it is already running. This model is needed when links from one application to another are used. Whenever an application needs to be interrupted, the system takes care of all pending data. If there is information that needs to be saved, the system saves it automatically into a default place, and if there is a pending dialog, it is canceled. The user is not asked questions; all the operations needed to bring the application to the target state are done automatically. After switching applications in this way, the **Back** function does not lead to the previous state. Instead, it functions as if the user had manually entered the second application and navigated to the target state: it leads to the previous state in the second application's internal structure even though the user did not actually navigate through it.

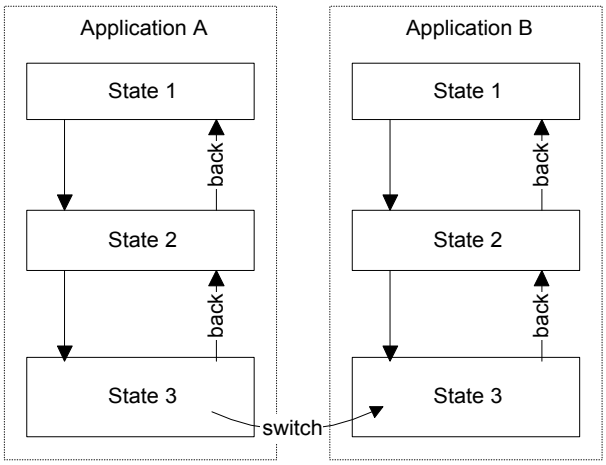


Figure 95: Switching between applications, interrupting application B. The Back function works inside application B, it does not lead back to application A.

8.4 Phone

Phone is the application for handling voice and video calls. It is a central application in the device. Calls can be created by dialing from the Idle state, and whenever there are one or more voice calls going on, the Phone application takes the place of the Idle state. Swapping between Phone and the Idle state is done via the **Options** menu or the Fast Application Swapping window.

In simple one-call cases, the Phone application looks as shown in Figure 96.



Figure 96: Phone application examples:
a) Creating an outgoing call
b) Active call going on
c) Incoming call alerting

More complex cases where more than one call is involved may also happen. The locations and sizes of individual call windows change according to the situation, see Figure 97.



Figure 97: Phone application, multi-call cases:
a) One call active, another on hold
b) One active, one held, one waiting call
c) User is entering a number while another call is active

When there are ongoing calls, but some other application than Phone is on top, the call windows are reduced into small pop-up windows in the top-right corner, see Figure 98.



Figure 98: Call status window, used when the Phone application does not reside on the screen

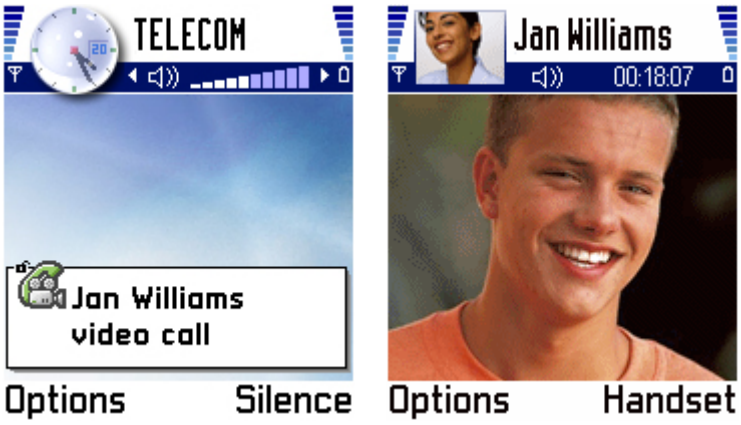


Figure 99: Arriving and active video call (from S60 2nd Edition, Feature Pack 2 onwards)

8.4.1 Call handling

Calls are handled using the Call creation and Call termination keys as follows:

Call creation	• Answers the incoming call when the phone rings or there is a waiting call. • Creates an outgoing call when there is a number entry window active. • Puts an active call on hold; un-holds a held call; swaps held and active calls.
Call termination	• Rejects an incoming call. • When there is an active call, ends the active call. • When there is only a held call, ends the held call. • When there is both an active and a held call, ends the active call and activates the held call.

The basic rule is that the call handling keys accomplish these same described functions regardless of whether the Phone application is on the screen or not, so the user need not swap applications for these operations when using another application (however, some applications may override the Call creation key and use it on a local function: e.g. in Messaging Centre, the Call creation key can be used to send a message).

8.4.2 Volume control

In case the hardware has no dedicated volume keys, the sound volume is adjusted using the Arrow left and Arrow right keys. The volume indicator is in the navi pane.



Figure 100: Volume indicator in the navi pane

It is worth noticing that the user must be in the Phone application (or another audio application) to be able to adjust the sound volume if dedicated volume keys do not exist.

The same volume adjustment method is used in other audio applications.

8.5 Viewers and Players

Viewers are used for displaying read-only data. Examples of viewers are SMS, Email, and Image Viewers.

To edit or create new data, the user starts an editor; the viewer may offer an option for starting an editor.

When a piece of read-only data is presented, no focus is needed, as the user does not need to access any individual object within the data; it is only necessary to be able to go through the data. Then the data is projected onto a virtual strip of paper that the user can move within the viewing window. There is no need to point at a specific part within the data, but a viewer may be able to zoom in and out.

8.5.1 Text viewers

In general, text viewers should follow these guidelines:

- Text is wrapped according to the width of the viewing pane. There should be no need to scroll horizontally while reading the text.
- Up and down scrolling should proceed page by page. A press of the Arrow down key displays the next screen of text (note that in editors, text is scrolled line by line).

8.5.2 Image viewers

A still image viewer is a tool optimized for viewing photographs or other images. A dedicated image viewer may offer features such as zooming and scrolling, full screen viewing, and adjustments of image parameters such as brightness and contrast.

By default, the image should initially be scaled so that it fills the available screen area.

Viewing functions are available in the viewer's **Options** menu. However, the Numeric keypad may also be used to control some of the functions; this allows quick access to frequently used functions, such as zooming.

8.5.3 Multimedia viewers

For viewers that support several forms of data, the capabilities should be extended while still keeping the core functions consistent with text viewers. For example, the viewed data may contain graphical still images that are displayed among text. The following additions to the earlier rules are applied:

- When necessary, an embedded image shall be scaled down so that it can fit the display as a whole. If the scaling capabilities are limited, a moderate amount of oversize can be allowed.
- When scrolling data, the user should be able to see each image completely. This means that at least one scrolling step should let the user view the full image (if scrolling would occur strictly one screen at a time, an image could be split into two parts, each one in a separate screen.)
- The user should able to view an embedded image better by opening it into an image viewer (note that focus may exist in order to select the desired image, although application designs without focus are also possible).

- After viewing an image in a separate image viewer, the user shall be able to come directly back to the multimedia viewer, in the position the user started from.
- In case of content that cannot be viewed, the content is replaced with an appropriate icon.

8.5.4 Audio and video players

Audio and video are time-based data. The players for such data need at least the following functions:

- Play
- Pause: Stops the player, but leaves the pointer in the current location so that playing can be continued later.

These commands can be mapped to the same key, as only one of them is needed at a time. These functions shall be accessible by a single key press. The positive action softkey is preferred for this purpose.

Additional functions may include:

- **Stop**: Ends playing the clip and places the pointer in the beginning of the current piece or section (audio or video clip).
- **Next piece**: Moves to the beginning of the next piece.
- **Previous piece**: Moves to the beginning of the previous piece.
- **Forward** / **Rewind**: Fast playing of the piece, using short audio clips, forward or reverse.
- **Faster** / **Slower**: Changes the speed of playing the audio without changing the pitch. Can be useful with voice memos, for example.

The playing functions shall be available in the player's **Options** menu. However, the numeric keypad may also be used to control some of the functions.

8.6 Text Editing

This section describes the common principles of text editing, focusing on Latin based scripts. Requirements of other writing systems and input methods, such as those used for Chinese or Arabic, are not covered in detail.

Character input is accomplished using the Numeric keys. The ITU-T standard for numeric keypads specifies the mapping of Latin (English) characters to the keys, assigning three or four letters to each Numeric key from 2 to 9. The input of a specific character can happen either by repeated key presses within a time-out, or using some language-specific algorithm that tries to find the correct characters according to the produced key sequence.

The other keys have specific functions assigned to them during text editing:

Numeric key 1	Punctuation. The most used punctuation characters are available using repeated key presses within a time-out.
Star (*) key	In the alpha mode, offers the next match in Predictive Text input. In the alpha mode, a long press opens the special characters table, where the complete set of punctuation and other characters

	are displayed. One character can be picked at a time. In a typical number mode, produces a * character, repeated presses produce other characters allowed within phone numbers: +, p, w
Hash (#) key	In the alpha mode, changes the character case (upper case, lower case, text case). In a typical number mode, produces a # character.
Clear key	Clears the previous character from the cursor. In case more characters are selected, clears all the selected characters.
Edit key	Press and release: opens the editing menu (see Section 8.6.1 below). Selects characters when pressed simultaneously with navigation keys (see Section 8.6.3 below).

8.6.1 Editing menu

Editing-specific functions can be accessed using the Edit menu. It is opened by pressing the Edit key when in a text editor. The editing menu looks and functions like the **Options** menu, but it contains only text editing functions whereas other available options remain accessible from the **Options** menu. The content of the editing menu is as follows (note that items that do not apply to the editor in use do not appear in the menu):

Predictive text options ▸		A submenu for Predictive Text options.
	Predictive text input on/off	A temporary setting of predictive input.
	Matches	
	Insert word	
	Edit word	
Alpha mode		Switch to the alpha input mode.
Numeric mode		Switch to the number input mode.
(Other input modes)		Other available input modes are listed as individual items in the menu. The other modes may be language-specific.
+ + +		
Cut		The Clipboard functions allow importing and exporting chunks of text.
Copy		
Paste		
Insert symbol		Opens the special characters table. The same action as a long press of the Star (*) key (see Figure 101 below).



Figure 101: Special characters table

8.6.2 Editing indicators

The status of the editor is displayed using graphic indicators. They inform the user about things such as:

- Editing mode (numeric/alphanumeric, language-specific modes)
- Character case
- Predictive Text status
- Available space

When the editor is in the main pane, the indicators are located in the navi pane. A Data query that resides in a pop-up window has its indicators within the window.

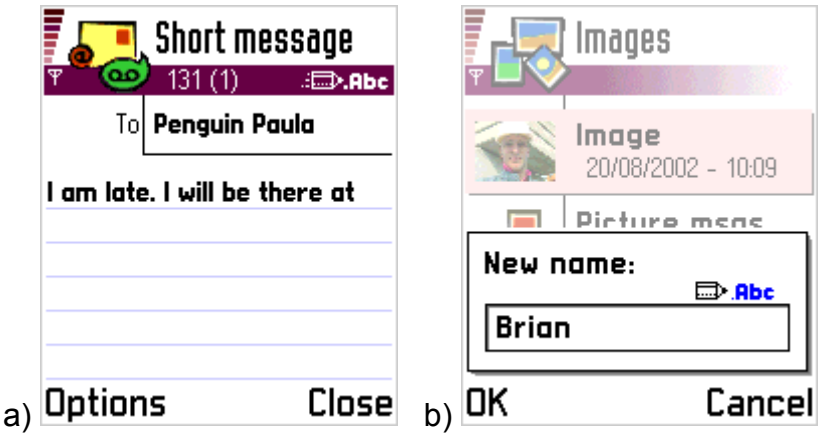


Figure 102: Editing indicators in the navi pane (a) and above the editor field in a pop-up window (b)

8.6.3 Selecting text

A chunk of text can be selected in an editor by keeping the Edit key pressed and using the navigation keys (Arrow left/right/up/down). The selected text is shown using highlight. The **Cut** and **Copy** functions are available when text is selected.

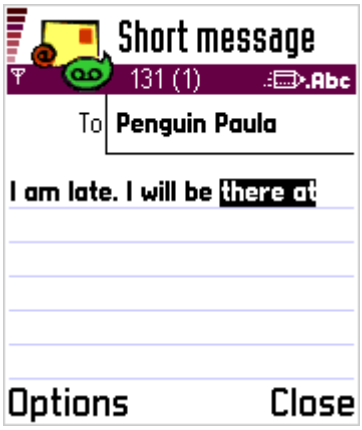


Figure 103: Text selection in an editor

If the user presses an Arrow key alone when there is a text selection, the selection disappears and the cursor reappears in the respective position.

8.7 Document Handling

8.7.1 Creating new documents

The user must be able to start typing (or otherwise creating) a new document without first being asked for a file name. In many cases, documents do not even have a name that would be seen by the user: short messages and notes, for example, are listed and managed by the beginning of the content.

Some document types have a file name the user can choose. Even then, the system may give a default name for the document, and the user is not forced to name it. When it is important that the user knows the document name, the following methods can be used:

- The system asks the user to confirm the name, using a query with the default name in place. This can happen at the moment the user is about to exit the editor, or when the user has selected the **Save** option.
- In some applications it makes sense that the user can give a default name in advance, and the system creates unique names from it by adding e.g. a number after the common name. This way it is possible to have meaningful document names without typing them for each file.

8.7.2 Saving edited data

Various kinds of documents may have different requirements concerning the behavior of keeping or discarding the edited data. Typically one enters or modifies data and then closes the editor, accepting the new data. But sometimes there may be a need to do something else, e.g. cancel the edit, or save an intermediate version of the data.

Simple queries and setting items offer the user one input element (an editor or a list), presented in a temporary window. The softkeys are **OK** and **Cancel**. The Selection key is assigned the same action as the left softkey: they both accept the input and close the query. The **Cancel** softkey discards all changes and returns. There is no need to ask for a confirmation from the user. The actions are clear, and the accepting option can always be seen in the softkey.

A form is a view with more than one input elements. The form element types are text (and numeric) fields, pop-up lists and sliders. A form may have separate View and Edit states; to go from the View state to the Edit state one has to select **Edit** from the **Options** menu. Most forms do not appear in the View state at all; the items are then always editable. In the Edit state, the user can move from one element to another and make changes.

Forms are somewhat different from queries and setting items as individual fields do not require an explicit accept action; one can move freely between fields and edit them at free will. The user can accept the whole form in one command.

In case there are no context-specific functions (other than acceptance and cancellation of the form), the interaction can be made similar to that of queries: the left softkey is **Done** and the right one **Cancel**. This arrangement is easy to understand.

However, often there are functions that must be accessible while editing the form, for example adding new items to the form. In that case, the **Options** softkey is needed, and the form cannot be accepted simply by using the left softkey. The solution to this problem is to assign the **Done** function to the right softkey. Pressing it accepts the data in the form and returns to the appropriate place. In case the **Cancel** function is needed too, it can be added into the **Options** menu, using a descriptive name such as **Discard changes**.

Document editors (such as a message editor) usually need the **Options** softkey. There may be message sending commands, preferences, help and other functions that need to be placed in the **Options** menu. In this case, too, the right softkey will save the data and return to the appropriate place; it is labeled as **Close**. A note telling where the data was saved should be given in case it is not obvious within the context.

In some applications there may be a need for discarding all changes, or saving intermediate versions of the document. These functions can be placed in the **Options** menu as required. It is worth noticing though that sometimes technical restrictions such as memory limitations, etc., may prevent for example discarding all changes to the document.

8.7.3 Folders

When there are a lot of data items to be managed, it makes sense to divide them into smaller sets. A folder is a place where a set of items can be collected. A folder can be present among single items in a directory, but it can be opened in order to view its contents. Users are able to create folders and delete them, move items into folders and out of folders, and rename folders and change other properties of folders (depending on the application).

The normal methods for managing folder are as follows:

- To create a new folder, the user selects the **New folder** option while in a data items list. The user may also create subfolders in certain applications.
- To add items into an existing folder, the user selects the **Move to folder** option while the focus is on the item to be moved. A list of the existing folders is offered, and the user can select the target folder. The softkeys are **Move** or **Open** and **Back**. The Selection key is assigned to the same action as the left softkey: **Move** when the focus is on the currently open folder, and **Open** when the focus is on a folder that can be opened.

8.7.4 Groups

Groups are another means of managing data in a container. However, unlike folders, groups do not contain actual data; there are only links to data that exist elsewhere. This makes it possible to access the same data from many places. A typical application for groups is a distribution list: one can collect a set of addresses into one group, to send messages to all the addresses by just referring to the group, and the same addresses can be present in any number of different groups.

Groups can be managed in much the same way as folders. There are some differences, however:

- Groups are presented in a separate Group view, not within the actual data items list. The Group view can be a tab view within the application.
- To create a new group, the user goes to the Group view and selects the option **New group**.
- To add items to a group, the user must be within the target Group view, and select the **Add items to group** option there. A list of items is then presented, typically as a multi-selection list, for the user to choose from.

8.7.5 Fetching data

Often it is necessary to be able to pick up a piece of data from an application, such as a phone number or address from Phonebook. This is called fetching. It is a read-only operation: the user cannot edit the data; only browsing and selecting are possible.

Browsing data during a fetch operation should resemble the application's normal use: the data should be arranged in the same way so that it is easy to find. Only the available functions are different: the data cannot be edited.

The left softkey during a fetch operation is labeled **Select** and it activates the same function as the Selection key: it selects the data currently in focus. The right softkey is **Cancel** and it returns to the previous state without bringing back any data.

Sometimes it is feasible to use a multi-selection list for fetching data. This should be done when it is probable that the user wants to select more than one item for fetching. An example of this is creating a group in Phonebook to be used as a distribution list: a list of names is offered to the user, and since the probable intention is to have more than one names in the group, a multi-selection list is a good tool to be used (see Section 6.2.4).

8.7.6 Memory card in data handling

If the phone and the application support memory card, the user is able to select the used memory when doing e.g. saving or moving operations. Usually this Memory selection List query is shown right after the user has selected to perform an operation.

In some applications there may be a need for a setting that defines the used memory in advance.

8.7.7 Embedded links

It is possible to initiate the downloading of content (e.g. images, tones, videos and applications) directly from certain applications, such as Media Gallery.

Downloading can be initiated from the **Options** menu. After the **Download** option is selected, Browser is opened as embedded and a bookmark folder that has been predefined for the selected content type is opened.

The actual download links exist in Browser's bookmarks and they can be edited only there. Links are divided into folders according to their content.

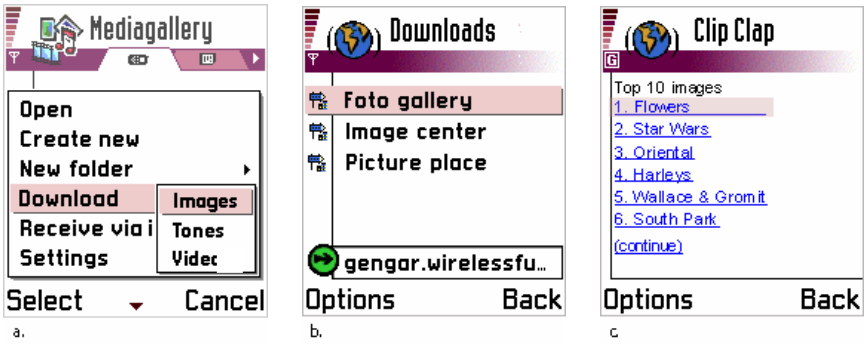


Figure 104: Example of an embedded link

8.8 Settings

Global settings, i.e. the settings that affect several applications or general issues within the device, are collected into the General Settings application.

Application-specific settings are handled within the application UI. They should be collected into a Settings view that can be accessed via the **Options** menu. The **Settings** option exists at least in the application's initial or main state, and possibly also in other states where it would be beneficial to have easy access to the settings, especially to certain context-dependent settings. The Settings view is a list of setting items (see the list item type description in Section 6.2).

Sometimes a general setting may be duplicated as an application-specific setting. The order of priorities for duplicate settings must be specified case by case. For example, a general Predictive Text setting could be overridden by an application-specific setting, but a general Silent mode setting within a profile should be effective regardless of any other tone settings.

If the number of setting items (within an application) is large, it may be necessary to divide them into groups. The grouping can be done in the following ways:

- Use tabs to access different setting groups.
- Design a hierarchical setting tree, use setting folders in the top level (there may be individual settings and folders in one list; this is an exception to the rule that only setting items should exist in one list).

8.9 Help

A device with S60 software may have a Help application containing information about the system, the functions it offers, and advice on getting the most out of it. The topics in Help are arranged by applications and in a hierarchical fashion so that they can be navigated in the usual way. The Help application is a normal application in regard to the ways it interacts with other applications.

Context Sensitive Help can be obtained by selecting the **Help** item from the **Options** menu within applications. The **Help** menu item opens the corresponding topic in the Help application. The user can further navigate the Help application to find more information.

9. Evaluate This Document

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