FORUM NOKIA

Symbian^3 UI Style Guide

Version 1.0; October 7, 2010

NOKIA

Contents

Change history

October 7, 2010	Version 1.0	Initial document release

1 About this document

1.1 Purpose

This document gives an overview of the Nokia Symbian^3 (S^3) 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 provides background material to help user interface designers make decisions about their products.

The style guide gives an overview and provides guidelines for designing good applications, but not all the information required 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 related documentation. There is no general discussion about good usability; instead this document clarifies how the style elements of the S^3 user interface are to be used in practice. For information and guidance on localisation and internationalisation, see reference document *[1]*.

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

Note: the illustrations in this document are examples only and are not necessarily pixel perfect renditions of the actual user interface.

1.2 Audience

This document is intended primarily for people who work on application design for the S^3 platform.

The document also helps persons such as product managers to get a general view of the S^3 user interface, what it is, and how it is intended to be used.

2 Changes to previous release

The major UI style changes to S^3 release from S60 5.0 release are:

- Supported display resolutions updated
- Renewed landscape layout introduced
- Renewed and optimised touch interaction introduced
- Sensor interaction defined more accurately
- Zooming interaction defined more accurately
- Task switcher updated
- New component, discreet pop-up, introduced
- List stretching introduced
- Home screen interaction and widgets
- Hardware variant updates

3 Where the Symbian^3 user interface belongs

The Symbian^3 user interface is used in mobile phones featuring personal information management (PIM) and multimedia applications such as:

- Telephony
- Calendars
- Text, multimedia, and email messaging
- Mobile browsers
- Imaging and audio applications and features

With standard ITU-T keypad devices, 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 at power users and require pressing two keys simultaneously.

The S^3 UI is also used for touchscreen devices, in which case the emphasis is on the fluency of task flows: enabling direct manipulation of the content, objects, and features, as well as minimising the need to switch from using a stylus or a finger on the touchscreen to using any of the hardware keys, is essential. Therefore, designs should strive to enable completing a task using the same interaction method it was started with.

4 Hardware

The S^3 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.

4.1 Display

The S^3 user interface display specifications are as follows:

- The S^3 common UI components have ready-made scalability, meaning that they can be adjusted to different display resolutions and orientations.
  Supported resolutions:
  - 640 x 360 (nHD) portrait and landscape orientations
- Square pixels
- Color capability (4096 or more colors preferred)

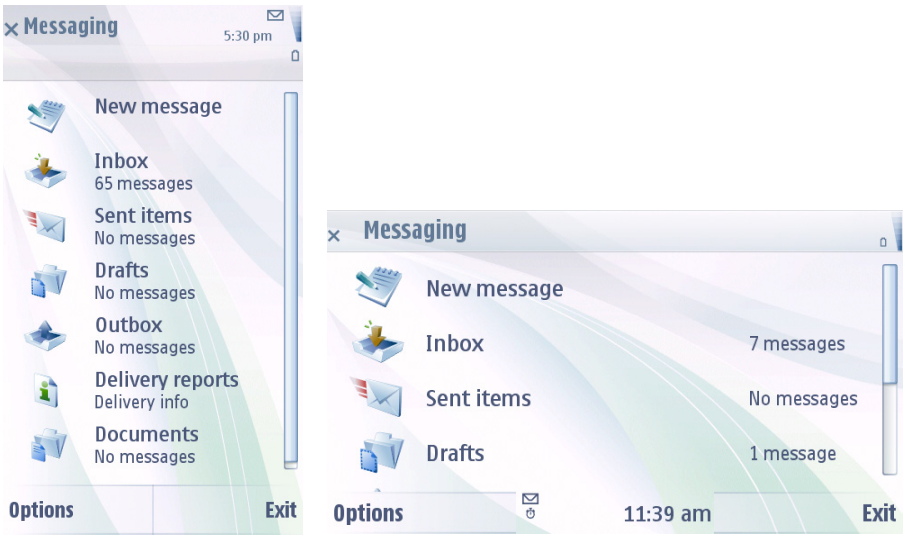


Figure 1 nHD Layout in portrait and in landscape

Both S^3 screen resolutions support touch interaction. The nHD layout is fully optimised for finger touch. Softkey labels are in the bottom of the screen in all supported resolutions.

For more information on touch-enabled devices, see chapter S^3 with touch in this document.

4.2 Keys

4.2.1 Required keys

The following table defines the required keys for the S^3 user interface.

Table 1 Mandatory keys for the S^3 nHD resolution

Keys	Description
Call handling keys	Call creation key (Send)
	Call termination key (End)
Home / Menu key	Access to home screen and menu

4.2.2 Recommended keys

The following table defines the recommended keys for the S^3 user interface.

Table 2 Recommended keys for the S^3 nHD resolution

Keys	Description
Other keys	Volume keys
	Power key
	Camera shutter key
	Keyguard switch

4.2.3 Optional keys

The following table defines the optional keys for the S^3 user interface.

Table 3 Optional keys for the S^3 nHD resolution

Keys	Description
Navigation keys	Scroll up, down, right, left, and Selection key
Other keys	Alphanumeric keypad, ITU-T, QWERTY, half QWERTY
	Voice key
	Zoom keys
	Other product or application specific keys

See section Keypad functions in this document for more information about the usage of keys.

4.2.4 Key definitions

The following recommended or optional keys can be added to a product to emphasise or facilitate some functions. These can be used to control applications or hardware such as spoken commands, sound recording, and audio volume control.

Table 4 Possible extra keys in the S^3 UI

Key	Description
Camera key	Used for turning the camera on and for taking pictures
Edit key	Used for launching the editing menu and for selecting text in editors
Power key	Used for turning the device on and off and for accessing the Profiles list (for quickly switching the active profile)
Voice key	Used for voice dialling
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) which 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.
Zoom keys	For zooming content, for example in the camera viewfinder or in an image viewer
Media keys: **Play/Pause** **Stop** **Rewind** **Forward**	Dedicated keys for controlling media (audio and video)

- 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 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 Arrow keys is pressed while the Edit key is held down.

4.2.5 Full QWERTY keyboard

The S^3 platform also supports devices with a full QWERTY keyboard. See section Typical full QWERTY keyboard functions in this document for details on keyboard usage. Regional keyboard layout variants (for example, Japanese) are also supported.

Figure 2 Example of an S^3 full QWERTY keyboard layout

4.2.6 Half QWERTY keyboard

In addition to the full QWERTY keyboard, S^3 platform also supports a half QWERTY keyboard. As opposed to full QWERTY, there are two letters mapped with one key in the keyboard. The user can enter characters by multitapping half QWERTY keys or by using predictive text input. Numbers and special characters can be entered by first pressing the Fn key and then a QWERTY character key.

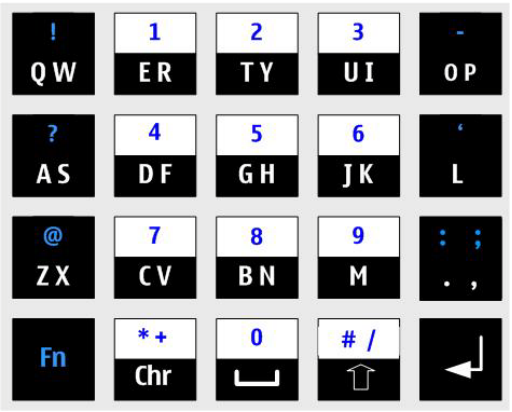


Figure 3 Example of an S^3 half QWERTY keyboard layout

4.2.7 Locking the keys and the screen

The keypad and touchscreen can be locked manually, for instance with a hardware key combination in a certain sequence (SK1 + Selection key) or via the Power menu. Locking may also be done automatically after a set inactivity timeout.

For touchscreens, it is recommended to have a mechanical keyguard switch with one position. In this type of mechanical switch, switching and releasing the key returns it to its original state. Another option is to have a mechanical switch with two positions (open and closed state).

4.3 Memory

One or several memory cards can be added to an S^3 product to increase the available memory. A device may also have internal non-removable memory or internal mass storage, supporting multiple drives.

The user is able to select the memory used when doing operations such as saving or moving files. 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 memory to be used in advance.

4.4 Acceleration sensors

S^3 devices support the use of acceleration sensors for movement-triggered functions. These sensors can recognise device orientation (landscape/portrait and screen down/screen up) and tapping (giving the device light taps with your hand).

4.5 Light sensors

Light sensors can be used, for instance, to switch keypad lights on and off according to environment illumination.

4.6 Proximity sensors

Proximity sensors are able to determine when there is an object near or against the front face of the phone. This data can be used, for example, to switch off the screen touch sensitivity during a call to prevent accidental touch-triggered events from occurring.

5 Graphical components

5.1 Windows and panes

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

Table 5 Windows and panes

Window / Pane	Description
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 subcomponent 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: • Main pane • Status pane • Control pane See sections Main pane, Status pane, and Control pane for more detailed descriptions on each of these panes.
Pop-up window	A pop-up window must 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. Backstepping should not usually lead to a pop-up window. Detailed information on the various pop-up windows can be found in section Pop-up windows.

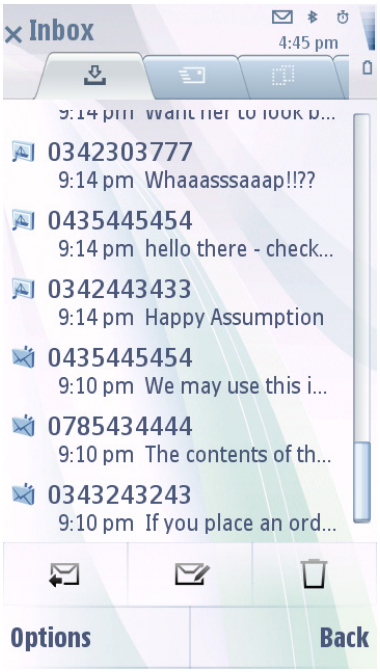


Figure 4 Panes

5.1.1 Standard panes in application windows

5.1.1.1 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.

Table 6 Standard components for applications to use in the Main pane

Pane	Description
List	There are several standard list types to choose from (see section Lists and grids for a detailed description of the list types). In landscape mode, two-row lists are most commonly stretched to be shown in one row so that more items can be displayed. The list stretching can also be disabled.
Grid	Items can be presented on a two-directional grid.
Find pane	The Find pane is used together with a list, and it allows the user to search list items alphabetically (see section Find pane for more information about the Find pane).
Status indicators	Status indicators are displayed in the Universal indicator pane. See section Universal indicators for more details on status indicators.
Soft indicators	Soft indicators only exist in the **Idle** state. See section Soft indicators for more information on status indicators and soft indicators.

Applications can also use the Main pane area to freely draw whatever is needed. In this case, however, the application's designer is entirely responsible for the look and feel of the Main pane. General guidelines for designing application-specific Main pane layouts can be found in chapter UI components.

5.1.1.2 Status pane

The Status pane displays status information of the current application and state, as well as general information about the device status, such as signal strength and battery charge. It occupies the top part of the screen and comes in two variants: extended and small. The Status pane may also be completely left out in certain applications or situations.

There are three different variations of the Status pane:

- Normal Status pane (Includes title and Navi pane)
- Flat Status pane (Includes title but no Navi pane)
- Thin Status pane (special Status pane used, for example, in the Browser application)

Flat and thin Status panes can be used in cases where it is imperative to maximise the Main pane area (for example, in the Browser).

The Status pane contains the following sub-panes:

- Title pane
- Navi pane
- Signal pane
- Battery pane
- Universal indicator pane

- Digital clock

Note that in landscape layout, the Universal indicators and Digital clock are placed in the Control pane area between the softkey labels instead of the Status pane.

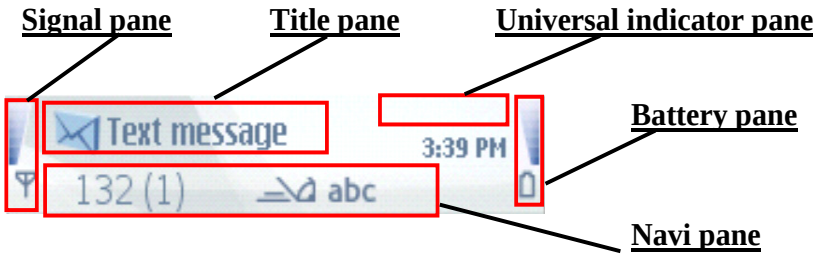


Figure 5 Status pane areas

See the following subsections for further details on each sub-pane.

5.1.1.3 Title pane

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

- Typically, the title text is the descriptive name of the current Main pane view.
- In an application main view, the title text is typically the application's name.
- In the **Idle** state, the title pane may contain a graphical operator logo (left-aligned in the area available) instead of the operator's name in text format.

5.1.1.4 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.

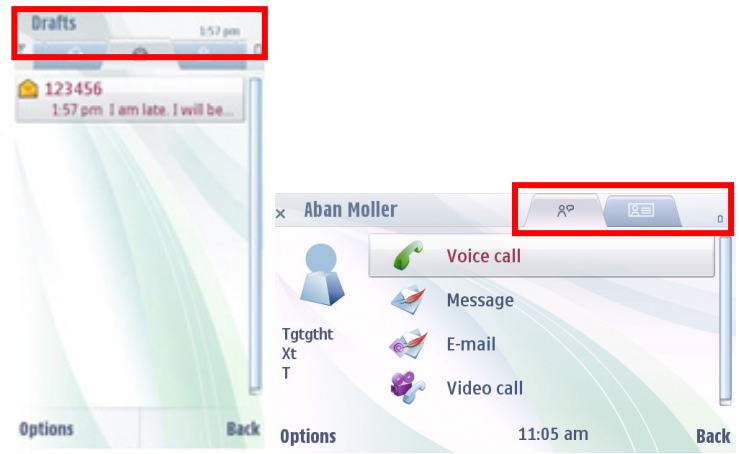


Figure 6 Navi pane

Depending on the context, the Navi pane can alternatively contain elements presented in the following table.

Table 7 Possible Navi pane content

Component	Description
Tabs 	Tabs are used when there are parallel data views that can be viewed alternately. Tabs can contain only icons and there can be a maximum of three tabs visible at the same time. The active tab is always centralised unless it is the first or last tab. As a design guideline, the number of tabs should be kept small (a maximum of six tabs is recommended), and the number should not be dynamic (see section Navigation using tabs for a description of the effect of tabs on navigation within an application).
Navigation text (icon+text) 	A navigation text is displayed in the Navi pane when there are similar items to be browsed by scrolling horizontally, for example, dates in Calendar. If textual tabs are needed, the navigation text should be used. Arrow indicators at both ends of the pane indicate the possibility to scroll.
Indicators 	In editors, the Navi pane contains editing indicators (see section Slider pop-up for a more detailed description of the Navi pane indicators).
Folder structure indication 	Opened subfolders are indicated in the Navi pane.
Application-specific content	When none of the above content types is suitable, the Navi pane content can be designed specifically for an application.
Empty pane	The Navi pane can be empty.

5.1.1.5 Signal pane

The Signal pane displays the cellular signal strength indicator.

Figure 7 Signal pane (left)

The indicator may also contain information about Packet Data (for example, GPRS) connection status.

5.1.1.6 Battery pane

The Battery pane displays the remaining energy level of the battery, using a graphical indicator. It also acts as a charging indicator. The battery indicator is visible in all applications.

Figure 8 Battery pane

5.1.1.7 Universal indicator pane and digital clock

The Universal indicator pane is used for displaying status indicators that need to be visible regardless of the current application. There is room for at least three indicators, although depending on the resolution, it may be possible to display more than three. The items are prioritised according to their importance.

In the landscape layout, the universal indicators are placed inside the Control pane area.

Figure 9 Universal indicator pane in portrait

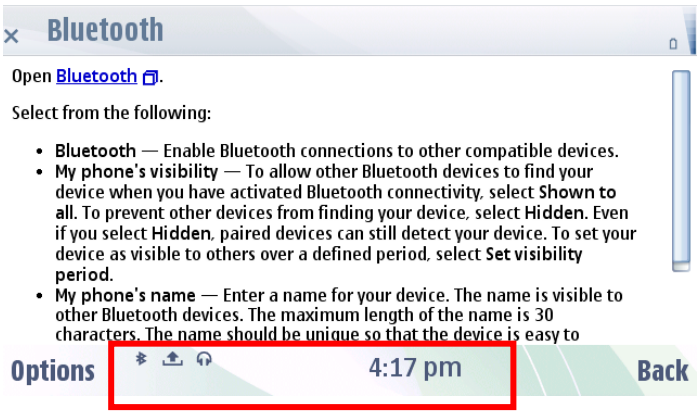


Figure 10 Universal indicator pane and digital clock in landscape

Tapping the Universal indicator and clock pane opens a Universal indicator pop-up with more information on Status pane indicators. For more details of the component, see Universal indicator pop-up.

5.1.1.8 Control pane

The Control pane occupies the bottom part of the screen and displays the softkeys with labels. In landscape, the Control pane is split into two and between the softkeys are the universal indicators and digital clock.

Figure 11 Control pane in landscape

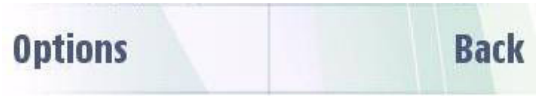


Figure 12 Control pane in portrait

The Control pane is also embedded in the pop-ups (including the Options menu). The width of the Control pane is the same as the pop-up.

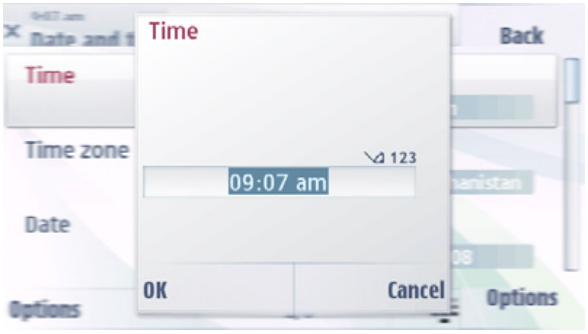


Figure 13 Control pane inside the pop-up

5.1.1.9 Scroll pane

With any list, grid, or other component that can be scrolled vertically, a scrollbar appears on the right-hand side of the component. The scrollbar is also displayed in pop-up components.

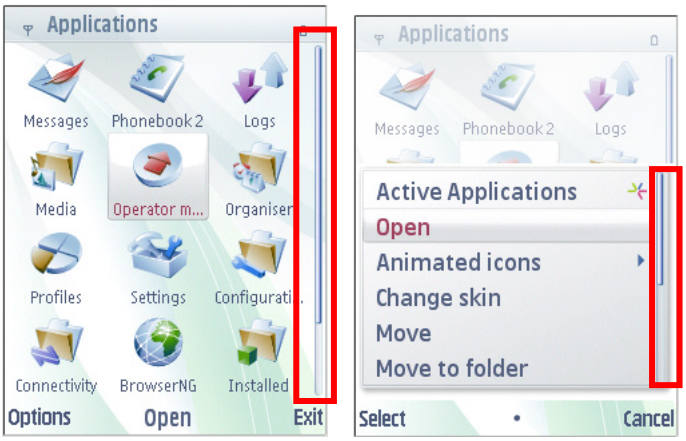


Figure 14 Scrollbar in the Main pane grid (left) and in a pop-up window (right)

- The placement of the scroll handle on the scrollbar reflects the position of focus in 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 scrollbar 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. The only exception is the Application menu, where the scrollbar is not displayed if all items fit in the same view.

5.1.2 Layout changes in bi-directional languages

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

1. The Status pane components are mirrored, for instance, the Signal and Battery panes change their places.
2. The Main pane lists and grids are mirrored: the A column icons and texts are on the right rather than on the left, and so on.
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.

5.1.3 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 backstepping from one state to the previous state does not lead into pop-up windows; they are skipped.

If the pop-up does not have softkeys, the background is not dimmed and the pop-up can be closed via touch down and release outside the pop-up.

More information on these components can be found in chapterUI components .

Table 8 Pop-up components

Component	Description
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 by 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, and 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. They contain text and possibly a graphical element, and their layout is similar to a **Confirmation** query. However, the softkey labels are typically non-existent as notes do not require user input and by default they disappear within a timeout.

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.
Discreet pop-up	A discreet pop-up is a temporary pop-up that appears on the upper left corner for a certain time-out. When a discreet pop-up appears, it does not dim the background and it does not have focus. The user can navigate in the underlying application normally. The pop-up can have simple touch functionality. It can be used, for instance, for information and confirmation note purposes.
Universal indicator pop-up	A universal indicator pop-up is opened from the Universal indicator pane. It contains more information of the active status indicators.
Call window	Incoming calls and outgoing calls are presented in pop-up windows. See section Call handling for more detailed information on call windows.

5.2 Zooming

5.2.1 Zooming types

Actual zooming of content applies to some texts, for instance, message viewers, editors, and some list components. There are two types of zooming, depending on the type of content:

- Text size adjustment should be available for text content without a strict layout, for example in email. When text size is changed, the text will be rewrapped to fit the width of the view. Seeing all content may require more vertical scrolling, but the intention is to avoid the need to scroll horizontally. The number of available text size levels may differ from application to application.
- With content that does not allow reformatting, such as images, maps, PDF documents, and many web pages, zooming changes the size of all content, keeping the aspect ratio of the content unchanged. Since the layout does not change, zooming may lead to a need to scroll both vertically and horizontally to see all content.

Whenever an application starts, the text/content size should be the default as defined in the settings. Each zoom adjustment the user makes is temporary, meaning that it will be discarded when the application exits.

5.2.2 Zooming interaction

The user should have a consistent means of producing zoom in and zoom out commands, but the means may differ from product to product. There are several potential zoom interactions depending on product features and form factor; each product should support the appropriate ones:

- In products with an ITU-T keypad, the * key zooms in and the # key zooms out. This functionality is not available in editors or other states where those keys have another role. In these cases the zoom functions, if applicable, must be available in the Options menu.
- In products with QWERTY, as well as half-QWERTY, the suggestion is to use the two topmost keys on the left hand side of the keyboard for zooming in and out. For example, in Latin keyboards the Q key would be used for zooming in and the A key for zooming out. Zooming symbols printed onto letters of the keyboard help the end user to find the feature. Like above, this zooming in/out feature is not available in editors.

- Touch stripes may be used to zoom. Finger movement along the stripe controls zooming. Applications may support different amounts of zoom levels; it should be possible to zoom across the full range with one swipe.
- Pinch zoom is available in touch products with multipoint-touch support.
- Some applications may support double tapping zoom. A double tap can toggle between zoomed and normal view, and there may be application-dependent behaviour.

5.2.3 Zooming feedback

Zooming should occur instantaneously, so that seeing the content change size would give the necessary feedback. A zoom indicator should not be needed in many cases.

However, an indicator can be provided when it helps the user understand what the current level among the available levels is. For instance, if there are three text sizes and the largest is already in use, it is useful to see an indicator of this when trying to zoom in. Such an indicator should be a floating component, and disappear after a short timeout.

5.3 Presentation of text

5.3.1 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, for instance, soft indicators in the **Idle** state, which are right-aligned.

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

5.3.2 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 an ellipsis (...) is 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 digits 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 components might be, for instance, queries to select an email address.

5.3.3 Marquee scrolling

When a text does not fit in a list item, the item text can be horizontally scrolling. This 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 in 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. The text scrolls until the end of it becomes visible, stops for a short while, and then instantly returns to the original position. By default, there is no looping or repeat.

Marquee scrolling is disabled when moving with the stylus.

5.4 UI themes

A UI theme is a customised graphical user interface (GUI) that replaces the phone's default look, and it is used to change the appearance of the user interface for the phone's applications. A phone with the S^3 user interface may provide several UI themes of which the user can activate one theme at a time. By activating a UI theme, the visual appearance of the user interface is changed. Note that this setting only affects visual elements, meaning that functions, the interaction style, or language settings are not affected.

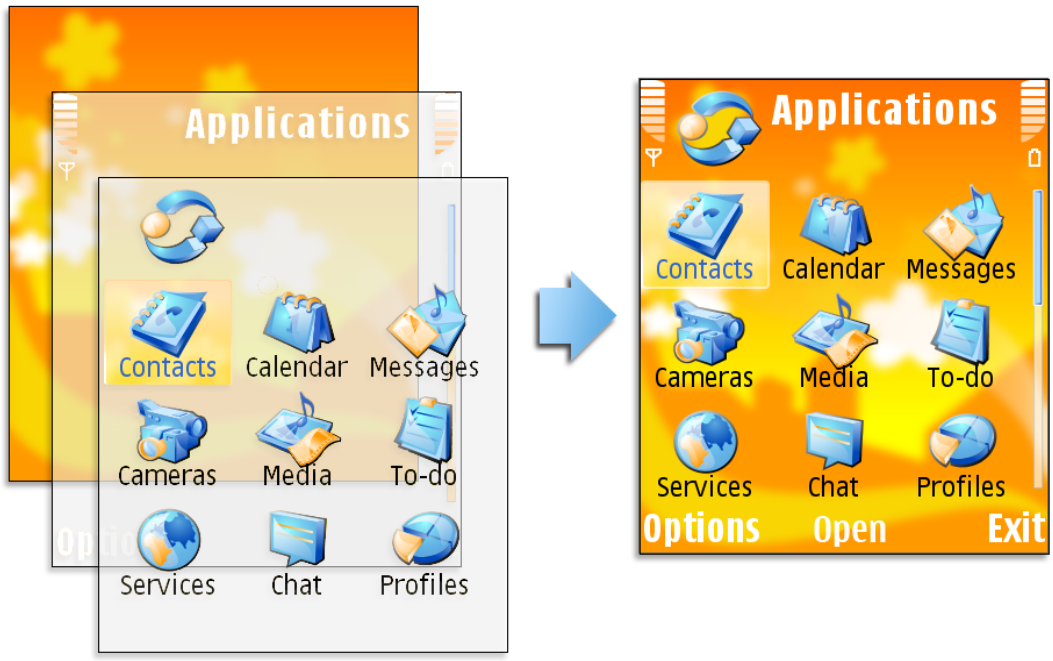


Figure 15 Simplified illustration of the UI theme concept

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

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

To ensure a consistent look and feel in all applications, it is recommended for the application UI designer to use the UI components that support UI themes. In case the application uses UI components that the UI theme does not support, the factory default of the missing items is used.

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.

5.5 Dimming

5.5.1 Dimming the background

The entire screen is dimmed in the background when the Options menu and Power menu, notes, soft notifications, settings editors, queries, or other pop-ups with softkeys are displayed.

5.5.2 Dimming of controls

In some situations, it may be necessary to dim unavailable controls. This is done on the component level so that a control has a normal and a dimmed state (for instance, the toolbar button).

Dimming indicates that the control is temporarily not available, for example, when a function is not possible for the focused item, but is possible for other items in the view. Dimming should not be used if the user never sees the control in the active state within the view. In this case, the unavailable control is to be hidden.

5.5.3 Dimming the Options menu items

The Options menu items are never dimmed. Instead, unavailable commands are either simply not shown or an Information note is given explaining why that item cannot be used at the moment.

6 S^3 with touch

This chapter describes the components and the interaction style for S^3 devices with a touchscreen. When designing an S^3 application, the UI must be designed so that everything can be done with touch, thus the design must not rely on hardware keys (except powering the phone on/off).

The touch UI in S^3 involves a number of specific touch components, component and UI behavior changes, layout changes, and additions to the interaction style. The main differences or additions include the following:

- Capacitive touchscreen (hardware)
- Toolbar in touch UI (component)
- Touch input (virtual keyboard, hand-writing recognition, and virtual ITU-T) (components)
- Stylus pop-up menu (component, finger-usable still)
- On-screen dialler (an application, see section Dialler)
- Additional touch functionalities to some components (for example, preview pop-up)
- Touch-enabled common components

6.1 Finger-usable UI

This document describes a finger as the input method with touch UI. Therefore, when using the term ‘stylus’, it actually refers to a finger as touch interaction method.

In S^3, the UI style defines the target minimum sizes for a UI element considered as finger-usable:

- 7 x 7 mm with 1 mm gaps for index finger usage
- 8 x 8 mm with 2 mm gaps for thumb usage
- List type of components should have minimum of 5 mm line spacing

The targets are general, as in practice the sizes can be use case-dependent, for instance due to frequency of use, efficiency vs. error criticality or ease of error correction, or the location of the button (edge of screen vs. center).

The visible area of the component and the component’s active area should be identical. There are exception cases to this rule though:

- When components are located near the edge of the display, the touchable area should extend fully to the edge of the display (in other words, beyond the component’s visible graphics).
- The visible area is smaller than the active area in order to keep the balance in the look and feel of the UI. For example, the scrollbar has a wider touch area than the visible area, for more information refer to sectionToolbar extension in touch UI.

- The active area is smaller than the visible area in order to avoid unwanted presses to contiguous active areas. In this case, there should be a graphical indication where the user should tap. Example: fixed toolbar buttons in landscape layout.

6.2 Touch interaction

In general, the touch UI emulates the functionality of the Arrow and Selection keys, but there are many deviations to this general principle, as described in the following sections. With good touch interaction design, users will choose to use the finger even if all the same tasks could be done on the hardware keys. On the whole, all S^3 common components are touch-enabled, allowing full finger-usage.

The touch UI in S^3 is designed to enable full task flow with touch, without the need to switch to hardware keys. Thus, the design ensures that users can complete a task with the chosen interaction method (keys or touch) from start to finish. The touch user interface offers some usability benefits compared to (or on top of) keyboard use.

There are a few good tips when designing applications for touch use. First, it is imperative that design decisions are based on real use cases. Secondly, as touch functions require a fair amount of 'discovery' from the user, it is good if only very obvious functions are made touch-enabled. In the same manner, it is good to keep in mind that not every item on the screen has to have a functionality attached to it and the selected functionalities should not be too surprising or radical. Note that theme design alone is not sufficient as a means of indicating touch functionality: where one theme may indicate touch, another may not.

Basic touch interaction is achieved through short tap and long tap. In a short tap, the finger is placed down and lifted up again on the same item within a short period of time. For example, an item can be selected from a list with short tap. A long tap performs some specific actions (described in more detail in section Touch down and hold) when the finger is held down on a component for a set amount of time. The following table lists the actions that take place on various movement types. The Drag and Drop functionality is not supported by S^3 as an integrated part of the style, but will be supported in some selected applications like moving items in the menu.

Table 9 Touch events

Touch type	Description
Touch down	The following actions take place on touch down: • The item is highlighted. • If there is a button under the stylus, it appears as pressed down. • Scrolling is activated if there is a scrollbar under the stylus.
Touch release	The following actions take place on touch release: • A menu command that was under the stylus is activated. • A button that was pressed down is released and the corresponding function is performed (unless it is a latching button as in a toolbar).
Touch down and cancel	Placing the stylus down on an object and then dragging the stylus before lifting the stylus cancels any action on the object. However, it is possible to drag the stylus outside the component and back onto it again, lifting the stylus up on the component, to perform the appropriate action.
Touch down and move	The user places the stylus on the screen and moves it on the screen without lifting the stylus up. This kind of dragging can be used in the following cases: • Scrolling lists and grids by moving the stylus over them • Selecting multiple objects in lists (see section Touch-based multiple

	selection) • Highlighting text in appropriate editor fields (i.e., 'painting') • Touch down and drag can also have application-specific usage, for example drawing lines or dragging selections.

6.2.1 Text fields

Tapping on a text field moves the text cursor to the tapped point. Blocks of text can be selected by moving the stylus while holding it down against the screen (dragging). Vertical dragging moves the content upwards/downwards as selecting multiple lines of text is activated by horizontal dragging that can then be continued with vertical movement.

6.2.2 Touch down and hold

Touch down and hold is used mainly to open the S^3 stylus pop-up menu. Applications can use an animation (starting 0.15 seconds after the start of a stylus down event) to indicate that a Touch down and hold action has been initiated.

6.2.3 Moving in touch UI

Moving in touch UI is done by touch down and move (dragging), touch down + move + touch release with speed (flick), or with the scrollbar.

In dragging interaction, the content follows the finger movements. In flicking interaction, the content continues its movement according to the parameters detected before the touch release. The movement of the content slows down with inertia.

When moving the content, the boundaries are indicated with an empty area appearing after the actual content. There can be as much empty area visible as the user can drag the display. Touch release hides the empty area immediately.

6.2.4 Interaction in hybrid devices

A device can have a combination of touch UI interaction and hardware key interaction. If the user is only using touch interaction, the highlight does not remain visible in the UI and there is no active focus. In hybrid devices, the user can provoke the active focus visible by pressing the Selection key or navigation keys. The first keypress activates the highlight and the next keypress initiates the action.

When the visible highlight is provoked to the Main pane with hardware keys, the item-specific options are updated to the Options menu and the hardware shortcuts work (for example, pressing the Send key to activate a call). For the navigation with Navi keys in the components, see Interaction style andUI components.

Visible focus disappears when the user starts to interact with the device using touch. When the highlight is activated again, the highlight comes to the same item from where it disappeared if the same window is still active.

6.2.5 Text input

There are several finger-optimised text input methods available for touch-enabled devices:

For portrait orientation:

- Virtual ITU-T keyboard with prediction

- Handwriting recognition (HWR) for some language variants, not available for Latin languages

For landscape orientation:

- Virtual QWERTY keyboard opened in full screen with input preview
- Handwriting recognition (when available)

If HWR is used in the product, the user is able to select between that and the virtual keyboard. This selection is recalled by the system until the end user changes the selection.

Figure 16 Virtual ITU-T keyboard

Figure 17 Full screen QWERTY keyboard

The virtual keyboard displays a keyboard that has a collection of basic keyboard functions (Shift, Caps lock, punctuation marks, and so on) and a text field where the user can see the given input without having to follow it from the editor field. Handwriting recognition allows the user to write characters on the screen with a finger.

If touch input has been launched from an editor, pressing any hardware character key closes the touch input. This applies also to a situation where the same product has both a hardware keyboard and virtual inputs available. Activating a hardware keyboard character key closes the virtual keyboard. Touch input never starts up automatically (a change of view does not launch it), but the user has to

launch it by tapping on the editing field. Note that merely moving the focus onto an editing field with the hardware keys does not launch touch input.

6.2.6 Adaptive Search

Adaptive Search is an application-independent Find pane-based filtering method for known content. The content may be presented in list or grid format.

The search strips the characters, symbols, and numbers from the items of known content, and displays them in a grid format on top of the content. The user is able to select them one by one and content is filtered to show matching items accordingly.

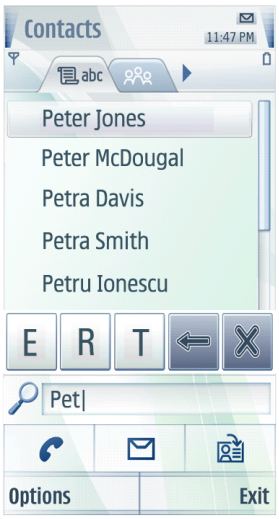


Figure 18 Adaptive search

6.2.7 Split view inputs

Virtual ITU-T and QWERTY can also be used in split view mode. Split view mode inputs are opened similarly to the full screen inputs. It is up to the application to use either the full screen input or split view input.

Split view input uses the application's own editor field and the application remains partly visible when the input is opened. Applications have the responsibility for keeping the editor visible while the user is typing. When the input opens, the Status pane shifts up and is hidden while the input is open. The split view is closed with the specific closing button in the input.

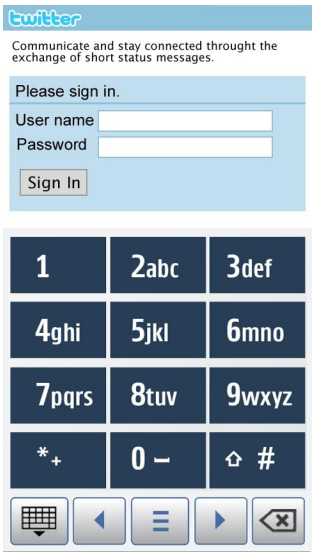


Figure 19 Split view ITU-T

Figure 20 Split view QWERTY

When the hardware keyboard is opened, the virtual keyboard (full screen or split view) disappears.

6.2.8 Touch-based multiple selection

6.2.8.1 Multiselection list

Items in a multiselection list can be selected with the stylus as well as with the keypad. A tap on an item in the list toggles the item between the marked and unmarked states.

If nothing is marked when the left softkey is pressed, nothing is selected.

6.2.8.2 Markable list

There are a number of tap + hardware key configurations for marking items in a markable list. These configurations depend on whether the device in question uses an ITU-T keypad with/without the Edit key, a full QWERTY keyboard, or no keypad at all. In the latter case, when there is only a stylus in use, marking is done via the Options menu by selecting **Mark/Unmark** or **Mark/Unmark all**.

A markable grid works in the same way as a markable list.

6.2.9 Cut-Copy-Paste

The **Cut**, **Copy**, and **Paste** functions are available from the Options menu in editors. In viewers, the **Copy** command is in a stylus pop-up menu that is opened via touch release after text selection. The dragging function works both in editors and viewers.

6.2.10 Preview pop-up in a touch interface

Preview pop-up is a floating component used for displaying more detailed data about an item that is touched or that has focus in the Main pane. It is not a touch-only component, and its basic functionality is described in section Preview pop-up. Although it is not recommended to place buttons or other items that appear touch-enabled into the preview pop-up, it is still possible to give some items touch-only functions. It is up to each application to decide whether to make use of this support in some sensible way. For example, a tap on a www hyperlink displayed in the preview pop-up would open the Browser and open the website in question. However, it should be noted that using touch-enabled items in the preview pop-up is an advanced, heavy user feature.

6.2.10.1 Preview pop-up for touch only

The previous section describes the general case for touch and the preview pop-up. There is, however, a variance possibility for the preview pop-up, which is to limit it strictly for touch. In this case, the guideline for having no buttons or other obvious touch items in the preview pop-up can be ignored for the most part: if the preview pop-up can only be opened with a stylus, it is perfectly acceptable to make touch functions plainly touch-enabled.

6.2.11 Discreet pop-up

Discreet pop-up is a component meant for showing notifications without blocking ongoing task flows in the UI. A discreet pop-up does not dim the background and the user can interact with the underlying application normally while the discreet pop-up is displayed. A discreet pop-up can contain an icon and a maximum of two rows of text. The pop-up can have simple touch interaction attached, for instance, tapping the pop-up can open an application.

The discreet pop-up can be used to replace the following note types: Error note, Warning note, Information note, and Confirmation note. The discreet pop-up does not replace wait notes, progress notes, or the soft notifications in the home screen. An application can have a soft notification on the home screen and use the discreet pop-up to display information of the event anywhere else in the UI.

Discreet pop-ups can be displayed in touch and non-touch devices, but in non-touch devices, the touch functionality of the component is disabled.

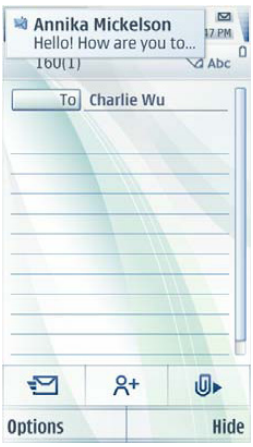


Figure 21 Discreet pop-up

6.2.12 Toolbar component in touch UI

The Toolbar component in general is described in section Toolbar. In touch UI, the toolbar is either fixed in the layout, or a floating component. Both toolbars cannot exist in one view.

Toolbars are view-specific, thus all views of an application do not have to have a toolbar. A toolbar can not contain any functions that relate to the Main pane item since there is no visible highlight.

Selection in touch toolbars is done with a single tap, and a related tooltip is shown with stylus down.

6.2.12.1 Floating toolbar in touch UI

The amount of floating toolbar buttons in nHD layout is three. In case a floating toolbar is used in full screen, there can be four buttons.

A floating toolbar can be opened and closed as in non-touch UI via the Options menu or with the Selection key, or it can be opened by tapping the Main pane area. In this case, closing is done by tapping the Main pane area again, with timeout or by selecting a function.

6.2.12.2 Fixed toolbar in Touch UI

The amount of fixed toolbar buttons in nHD layout is three. A fixed toolbar reserves its own space from the layout and does not cover the content of the Main pane area.

A fixed toolbar cannot be opened or closed, but it is always visible. The toolbar can be hidden in case other controls are also hidden, for example, in full screen views.

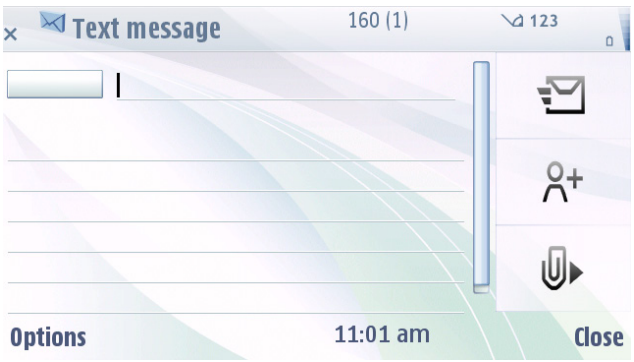


Figure 22 Toolbar in touch UI fixed in the layout

6.2.12.3 Toolbar extension in touch UI

As in non-touch UI, toolbars can also have extensions in touch UI. For a floating toolbar in touch UI, the same rules apply as defined for floating toolbar extension in general in section Floating toolbar extension. In touch UI, the toolbar extension can always be closed with the same toolbar button it was opened with, or by tapping outside the extension area.

In addition to general extension rules, the toolbar extension for a fixed toolbar has certain specific guidelines. Besides buttons, the extension of a fixed toolbar can include editor fields, for example, for entering an URL or for the **Find** function in the browser. The most important editor field is to be the topmost and it has the cursor by default.

Note: In case editor field(s) exists, there is less space for buttons in landscape layout compared to portrait layout.

6.2.13 Touch scrollbars

In S^3 touch devices, scrollbars (vertical and horizontal) have a touch-enabled scroll box. The hardware keys (Arrow keys) can also be used for scrolling when the scrollbars are visible. The scrolling movement on the screen is smooth.

In order for the scrollbar to be more usable with touch, the actual scrollbar area is wider than the visible scrollbar. When the user scrolls, the scrollbar has a related effect to indicate the scrolling (see Figure 23). Because the actual scrollbar area is wider, the items appearing on that area in a scrollable list, grids, viewers, and so on cannot be touch-enabled. For example, list icons in column D cannot be tapped.

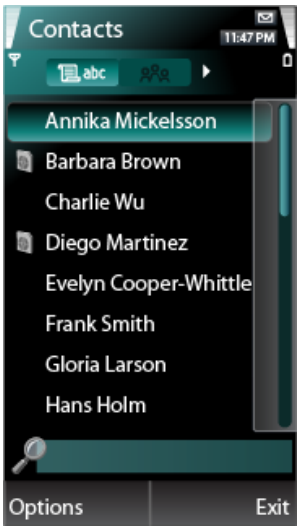


Figure 23 Touch scrollbar

6.2.14 Stylus pop-up menu

The stylus pop-up menu is a separate pop-up mainly used to show the item-specific functions related to the item it was launched from.

The activation of the stylus pop-up menu is usually done with long tap. The background is not dimmed when the stylus pop-up menu is shown. The menu disappears automatically after 10 seconds (or as soon as the user selects an item or taps outside the pop-up area). A highlight remains visible in the parent item as long as the menu is open.

As the purpose of the component is only to offer some primary functions (maximum of six functions), the stylus pop-up menu is not a scrollable component.

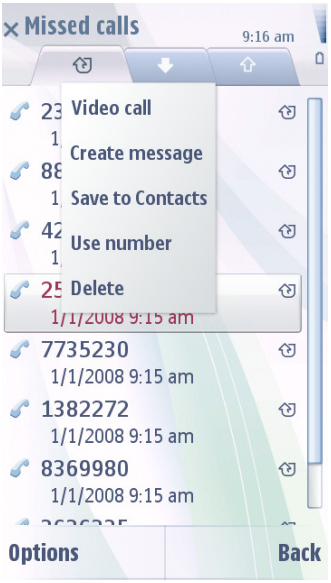


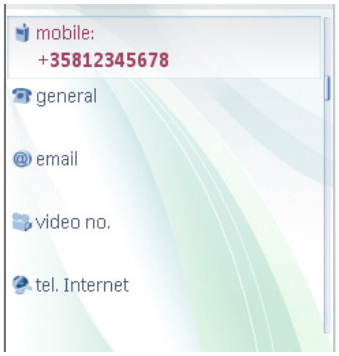
Figure 24 Stylus pop-up menu with item-specific functions

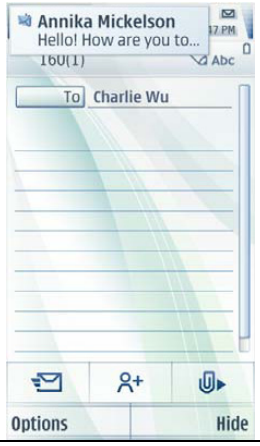
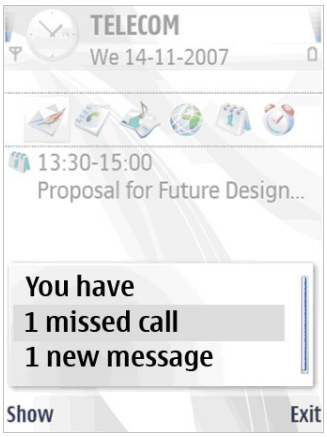
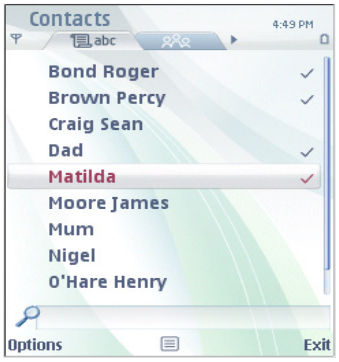
6.2.15 Universal indicator pop-up

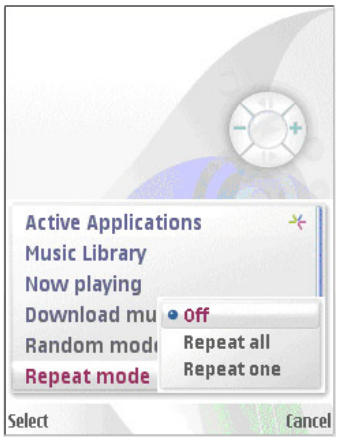
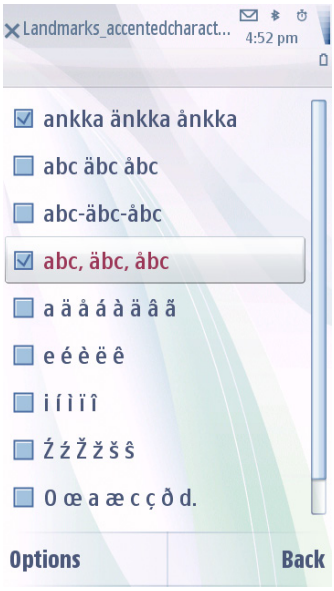
The universal indicator pop-up is a touch-only component. It is activated by tapping the universal indicators or the digital clock. It always includes battery and connectivity indicators and the digital clock. The user can tap any of those for more information. For each visible universal indicator, there is one row reserved in the pop-up for showing the icon and a description text. The description text may also contain information about the current status and/or it may be a link to a view in a corresponding application, for example, tapping the text '3 new messages' opens the relevant view from the messaging application, and tapping 'Bluetooth active' switches the view to the Bluetooth settings. The content of the row is determined by the application originating the indicator.

6.3 Touch support for common S^3 components

Table 10 Touch support for common S^3 components

Component	Example image	Touch functionality
Forms		In the **View** state (in a normal two-row list item), tapping on an item activates it. In the **Edit** state, the highlight is visible all the time but it works with single click. One tap moves the highlight and launches a touch input window. Within a focused item, the functionality is as for a text field.

Discreet pop-up		Touch down keeps the pop-up visible. Touch release either dismisses the pop-up or activates the application-specific functionality. Most commonly it activates the application.
Form items: • Text field • Pop-up field • Slider		As for equivalent setting items.
Grouped soft notification, **List** query, **Grid** query		A single tap activates the item.
Left/Right arrows		A non-focusable item. One tap to activate.
Markable list Selection list		One tap on a focused item activates it. The marking functionality is described in section Touch-based multiple selection.

Menu list (Options menu and submenus)		A single tap activates the item. If the item has a submenu, the highlight remains visible in the parent item.
Multi-selection list		One tap marks/unmarks the item. The marking functionality is described in section Touch-based multiple selection.
Preview pop-up		See section Preview pop-up in a touch interface.
Scrollbar		See section Toolbar extension in touch UI.
Setting items		
Pop-up setting		Pop-up setting: A single tap activates the item (emulates a Selection key press).
Multi-selection list setting		Multi-selection list setting: the functionality is as for a multi-selection list.
Text setting		Text setting: the cursor can be moved with the text field. Dragging selects a block of text.
Slider setting		Slider setting: the slider thumb is draggable.

Setting lists		One tap on a focused item activates it (emulates a Selection key press).
Softkeys (left and right)		A non-focusable item. One tap to activate.
Stylus pop-up menu		See section Stylus pop-up menu.
Tabs		A non-focusable item. One tap to activate.
Toolbar		See section Toolbar component in touch UI.
Slider pop-up		A draggable slider thumb. Icons can be set as tappable. See section Slider pop-up.
Volume Control		The volume slider thumb is draggable, and muting can be done directly by tapping the speaker icon. See section Volume Control pop-up.
Status pane • Signal pane • Title pane • Battery pane • Navi pane • Status/universal indicator pane		Non-focusable items. A single tap activates the functionality if the sub-pane has one (the Signal and Battery panes are inactive).
Universal indicator pop-up		Tapping the link performs the application-specific functionality, most commonly opening the application. Tapping the description text without the link functionality has no effect.

7 Interaction style

7.1 Keypad functions

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

7.1.1 Key presses

A key press is the 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 Keypad tone 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 a 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 Home / Menu 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 a key repeat. Key repeat starts after the 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 actions 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 (for instance, 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.

7.1.2 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 on or off by the user.

7.1.3 Typical functions of the standard keys

Table 11 Standard key functions

Keys	Functions
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. • Adjusts the sound volume during calls and sound playback when the device does not have dedicated volume keys.
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. • Changes the value in a pop-up field immediately in forms.
Selection key	• Opens the focused item (such as a document or a folder) in selection lists and grids. • Selects an option in menus and lists. • Opens the context-sensitive Options menu when there is no item to open and no option to select (see section Selection list). The Selection key must not directly activate any function the user would not expect in the given situation: the Selection key is strictly a Get-in or Accept key.

	Therefore, the context-sensitive Options menu shall be offered in states where no selectable items exist.
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. • Also functions as a shortcut for sending files. 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 Call handling for more detailed descriptions). In the **Idle** state: • Brings up the **Last Dialled Calls** list for redialling.
Call termination	• Rejects an incoming call. • Ends an active call. • When there is only a held call, ends the held call. • When there is both an active call 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. The application is terminated, except for applications that play something in the background such as music players or radio. Any unsaved data is automatically saved (also in case the user has pressed **Exit** in the Options menu). Additionally in devices that have a dedicated Power key: • A long press closes down all connections (for example, GPRS, data call); however, this has no effect on IR and Bluetooth (see section Call handling for more detailed descriptions). Additionally in devices without a dedicated Power key: • When control is in the **Idle** state (that is, there are no active calls), a short press opens the list of Profiles (the Power key menu) and a long press turns the device off regardless of whether there are active packet data connections in the background or not. • When in the **Idle** state without any packet data connections active in the background, a short press opens the Power key menu and a long press turns the device off.

Home / Menu key	• Opens the home screen. • When in the home screen, goes to the menu. • A long press of the key opens the task switcher, 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. • Deletes documents or other entities in lists and viewers (these functions always require confirmation from the user). The Clear key is not used for backstepping or exiting; it is only used for deletion.
Edit (optional)	• Opens the editing menu in editors; the menu contains functions for input mode changing and other editing functions. See section Editing menu for the contents of the menu. • In editors, can be used together with the Arrow keys to select (highlight) text, which then enables the **Copy** and **Cut** functions. • In markable lists, using the Edit key together with the Arrow keys allows the user to mark several items in the list. After this, 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 usual. This is to enable the **Mark/Select** function where the key is being held down as a modifier key (see sections Editing menu and UI components for more detailed descriptions of the **Select** and **Mark** functions.)
Volume keys	Adjust the sound volume during calls and sound playback.

7.1.4 Typical full QWERTY keyboard functions

The full QWERTY keyboard default short key press events are the following:

Table 12 Default short key press events in full QWERTY keyboard

Key	Action
Character keys	Add a character.
Backspace	Moves the cursor one step back, clearing any character or selected text in that position.
Enter	Adds a carriage return where possible (a new line). Elsewhere the default behaviour is similar to the Selection key.
Space	Adds a space.
Chr	Opens the special characters table.
Shift	Activates the Shift character mode for the character that is pressed after Shift. For example, consecutive presses of Shift and the character 'a' produce a capital 'A' character.
Scroll keys	Move the cursor or focus in the pressed direction.
Fn key (optional key)	Activates Fn-mode character input for the next following key press. Two presses activate Fn mode (locked) until a single press deactivates the mode.

The full QWERTY keyboard default long key press events are the following:

Table 13 Default long key press events in full QWERTY keyboard

Key	Action
Character keys	Add 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 or focus to the pressed direction with key repeat.

The full QWERTY keyboard key combinations are the following:

Table 14 Key combinations in full QWERTY keyboard

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 is required.
Chr + a character key	Accented characters can be entered by holding down the Chr key and then pressing a character key repeatedly.

'Dead' keys	Some keys in certain localised 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 + a scroll key).
Ctrl + 'c'	The **Copy** function in editing (for example, text that was selected with Shift + a scroll key).
Ctrl + 'v'	The **Paste** function in editing (for example, text that was selected with Shift + a scroll key).
Fn + a character key	Activates Fn mode for that key press.
Fn + space	Opens the editing menu.

For devices that contain both a touchscreen and a QWERTY keyboard, physical keys can have functions mapped with touchscreen interaction. For instance, pressing Shift / Ctrl key while tapping an item on the touchscreen marks that item in a list.

7.2 Interaction with other hardware components

In addition to traditional keys, interaction devices of other types may be built into mobile devices. In most cases, they produce similar input events as keys do, so that the software need not be changed – the applications do not care what kind of hardware is used to give commands.

Examples of specific interaction devices are optical joysticks, optical 9-way finger navigation, and acceleration sensors:

7.3 Optical joystick

An optical joystick may be used instead of (or in addition to) the traditional navigation keys.

When used together with traditional scroll keys, an optical joystick controls the same functions as the mechanical scroll keys. The speed of the movement may differ, however.

When used alone as the sole scrolling device, an optical joystick emulates the functions of mechanical scroll keys. Sliding a finger downward over the joystick provides the same function as pressing the down key. Repeat functionality may be produced with longer strokes.

7.3.1 Acceleration sensor

On the whole, there are three types of interaction events that can be recognised with acceleration sensors:

1. Tapping the device by hand (double tap) to perform a given function in set events. Tapping interaction is not recommended for touch devices in order to prevent the unintentional touch actions it may cause if tapped in the touchscreen area.
2. Changing the device's orientation from portrait to landscape and back to perform screen re-orientation.
3. Changing the device orientation from screen up (for example on a table) to screen down and back to silence an alert tone (for example, silencing an incoming call).

The system is capable of sensing any of these events. The supported sensor interactions are configurable per device, but they must be kept consistent, not assigning different functions to these standard events. Also the user can be given the option to set which one to use. For example, it is

possible to set methods 1 and 3 (double tap by hand and turning the screen downwards) for the Silence incoming call event. The settings are available in General Settings.

A double tap on the device produces tactile (vibrating) feedback in order to reinforce and acknowledge that the interaction has been successful.

7.3.2 Tactile feedback

There are two types of cases where vibration of the device is used as an output method:

- As a **tacticon** to inform the user that some important event is taking place in the device. For example, new message arrival is indicated with vibration, or Warning notes can provide vibrating feedback to address the need for the user's attention.
- As **tactile feedback** to indicate to the user that an interaction event has been successful. For example, tactile feedback is given when the user presses a button on the touchscreen.

As with sounds, tactile feedback must be used carefully so as not to desensitize the user to the vibration: the attention-grabbing quality remains and functions as long as the feedback is not too frequent.

Tactile feedback is included in all common UI components. When new components are designed, tactile feedback has to be included in them. For example, in any button type of UI component the tactile feedback is natural. An application can disable tactile feedback from the common UI components it uses, if required. This is acceptable only in cases where tactile feedback would cause interference.

The user can choose the level of tactile feedback and whether the tacticons are on or off.

7.4 Navigation

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

- Tabs
- S^3 key and menu
- Direct navigation between sibling folders
- Links to applications and documents

These features are described in the following sections.

7.4.1 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 25), the **Exit** function replaces **Back** in the right softkey and is used for closing the application.

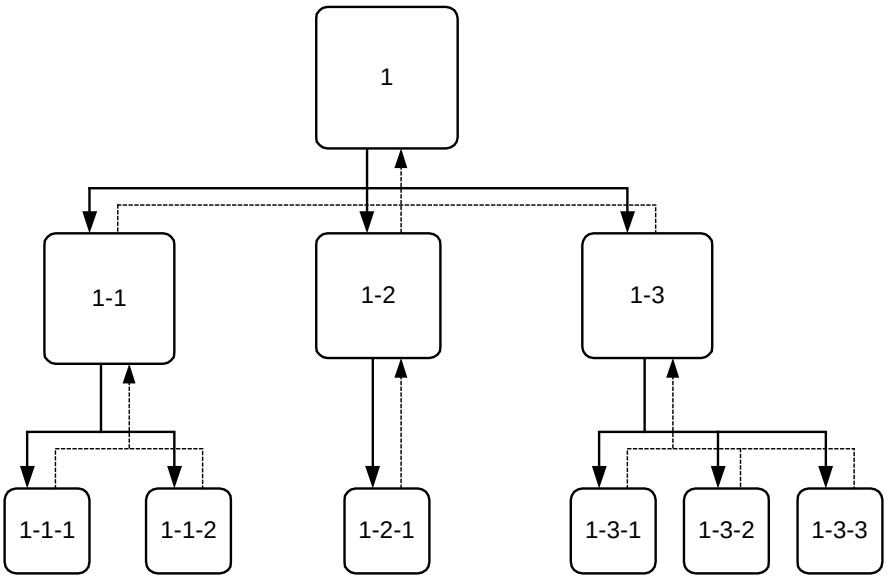


Figure 25Example of the basic state hierarchy in an application. The solid lines indicate moving forward from a state into a substate. The dotted lines are backward moves to the previous level.

7.4.2 Navigation using tabs

The S^3 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 by tapping some of the visible tabs as indicated in the Navi pane or pressing the Arrow key left and right.

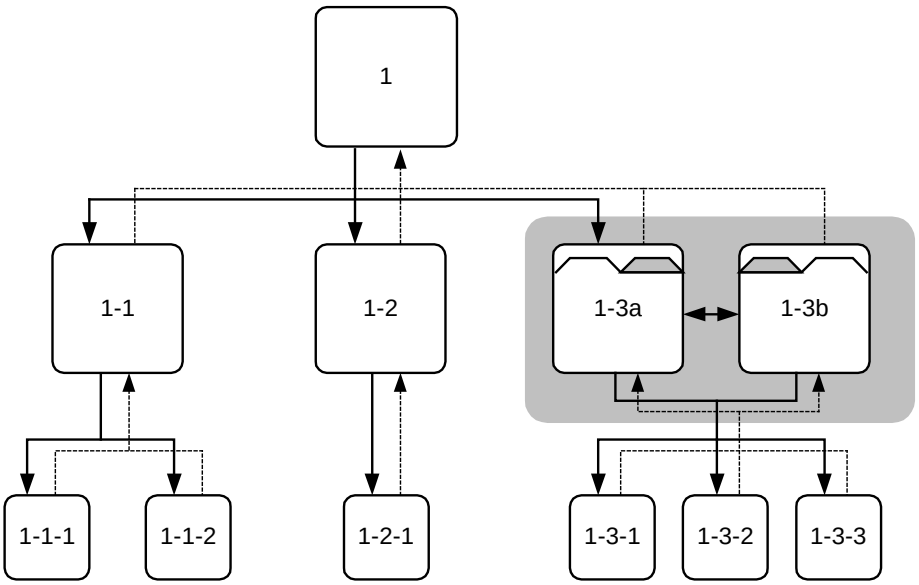


Figure 26In this example, state 1-3 uses two tabs to present its information. The user moves between views 1-3a and 1-3b using the Arrow left and Arrow right keys. Note that there is no **Back** function moving between the tab views; **Back** from both of them leads to state 1.

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.

7.4.3 Hierarchical application structures

When browsing within a hierarchical application structure, the S^3 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.

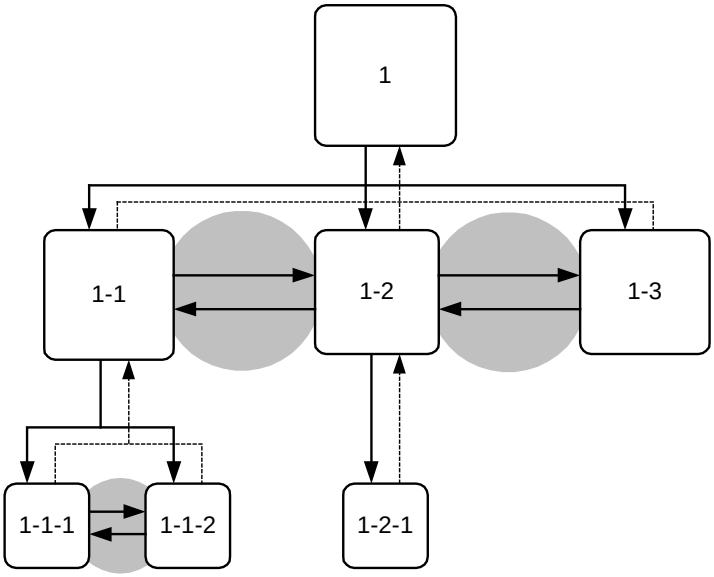


Figure 27In this example, each displayed state is a folder containing a list of items. State 1 contains three folders (1-1, 1-2, 1-3). When the user has opened one of these, there is a possibility to move directly into the other two folders without first going back to state 1. Folder 1-1 contains two 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.

7.4.4 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.

7.5 Multitasking

The S^3 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 an application without terminating another one that is presently being used, the user can either press the Home / Menu key to go directly to **Home** and from there to **Menu** and select the other application from there, or use the Task switcher.

The Task switcher shows the currently active applications. The most recent task is shown on the left. The second recent task is centered. Home screen is the leftmost item and it can't be closed.

Task switcher can be opened with / by:

- Long press of Home / Menu key
- Short press of dedicated Task switcher hardware key (if one exists)
- Selecting **Active applications** in the Options menu.

The user can either activate or close a task in the Task switcher. A task is activated by tapping it or by pressing the Selection key. Application is closed by pressing the **Close** badge on the upper right corner or pressing the hardware backspace key on the focused application. Touch down and hold on an item opens a stylus pop-up menu with the possibility to close one or all applications.

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

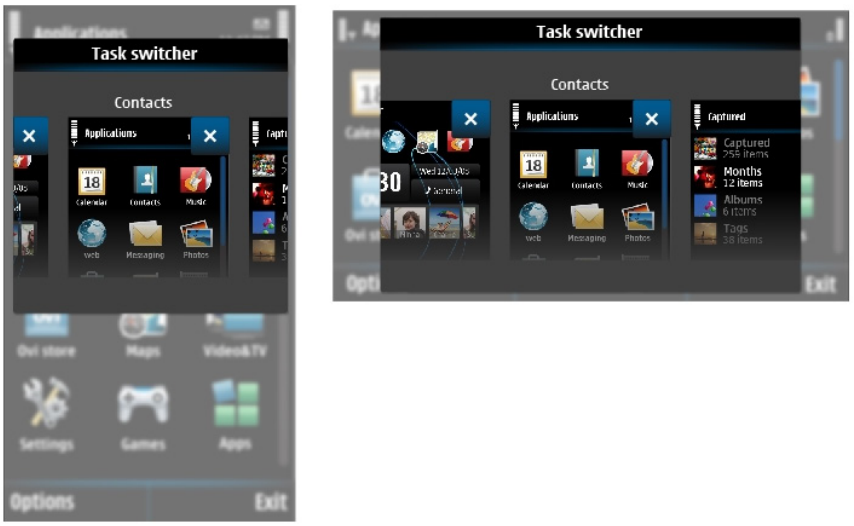


Figure 28 Task switcher

7.6 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. As described earlier, S^3 also supports the QWERTY keyboard. The input of a specific character can happen either by repeated key presses within a time-out, or by using some language-specific algorithm that tries to find the correct characters according to the produced key sequence.

Specific key functions (related, for instance, to the Star (*) or Hash (#) key) depend on the keyboard type used (ITU-T or QWERTY) and whether text prediction or multitapping mode is active instead. The four available modes are:

- ITU-T with multitapping mode
- ITU-T with predictive mode
- QWERTY keyboard with prediction on
- QWERTY keyboard with prediction off

7.6.1 Editing menu

Editing-specific functions can be accessed using the editing menu. It is opened by a short press of the Star (*) key (or by pressing the Edit key in devices with 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):

Table 15 Editing menu options

Special character row		One row with the most common special characters / symbols.
More symbols		Opens a full grid of special characters / symbols.
Predictive text ▸		Opens a submenu for the text prediction. Note: some devices may have more advanced text input and they have different options in use.
	Matches	Displays a list of matched words when predictive mode is on and a word is active in the editor.
	Insert word	Opens a query for inserting a word (no prediction).
	Edit word	In prediction mode, opens the active word from the editor in a query window for editing.
	Prediction on/off	A temporary setting of predictive input.
Alpha mode		Switches to the alpha input mode.
Number mode		Switches to the number input mode.
Copy text		Starts text copying.
Cut text		Starts text cutting.
Paste		Pastes cut or copied text in present position.
Writing language		Affects key mappings and the predictive input dictionary.

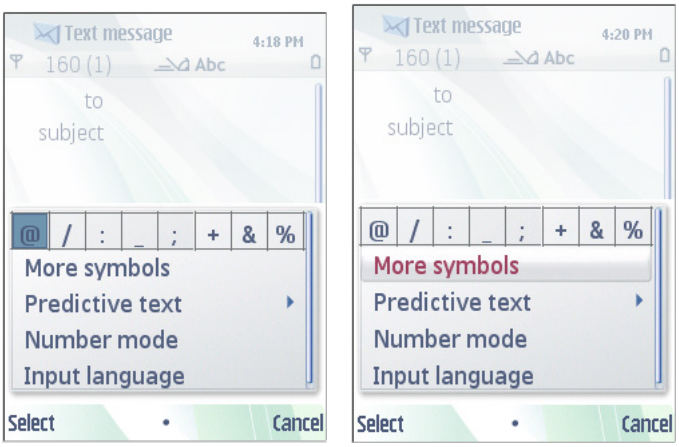


Figure 29 Editing menu

7.6.2 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.

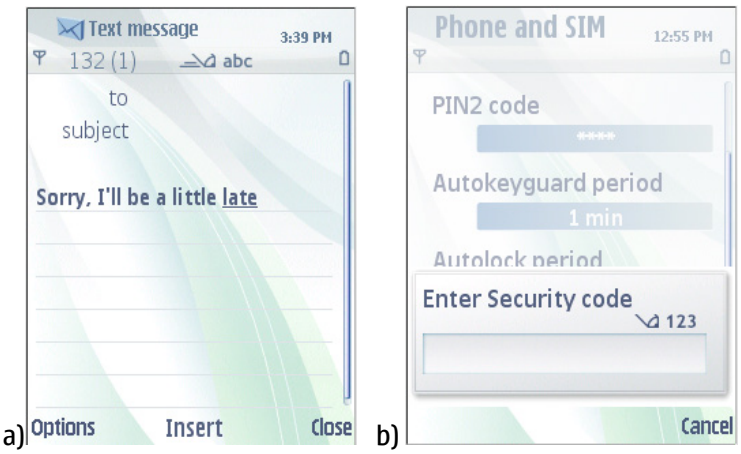


Figure 30Editing indicators a) in the Navi pane and b) above the editor field in a pop-up window

7.6.3 Selecting text

A longer stretch of text can be selected by keeping the Hash key (#) pressed while using the Arrow keys. The selected text is highlighted, and the **Cut** and **Copy** functions are available on the softkeys when text has been selected. It is also possible to start selections by going to the Options menu and selecting **Copy mode**. (Note: text selecting options in touch devices are described in section Cut-Copy-Paste)

The same function is done with the Edit key in devices that have one.

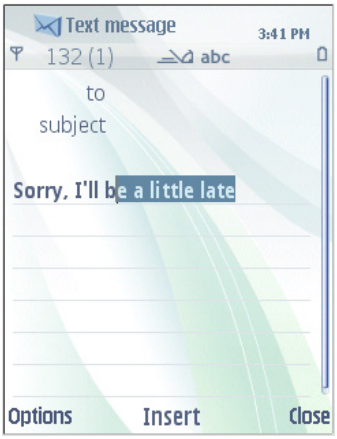


Figure 31 Text selection in an editor

If the user presses an Arrow key by itself when text has been selected, the selection disappears and the cursor reappears in the respective position.

8 UI components

This chapter describes the component features and interaction when using the navigation keys. For touch interaction, refer to S^3 with touch.

8.1 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 Arrow keys. This is called browsing or scrolling.

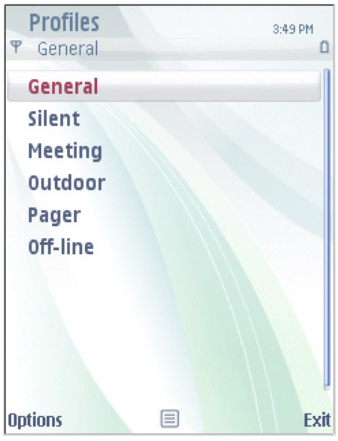
In general, vertical browsing is preferred over horizontal browsing; keypad solutions should take this into account.

8.1.1 Highlighting

When moving with the navigation keys in a list or a grid, one item in 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 (theme related) 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.

a) 

b) 

Figure 32 List highlighting (a) and grid highlighting (b)

8.1.2 Empty lists and grids

If there are no items to be displayed in a list or grid, the pane will contain text informing the user about the empty list. The information displayed in the list should be designed to provide as much help and guidance to the user as possible. Instead of merely stating that a list is empty, the text should point the user to some useful direction in the application in order to add content.

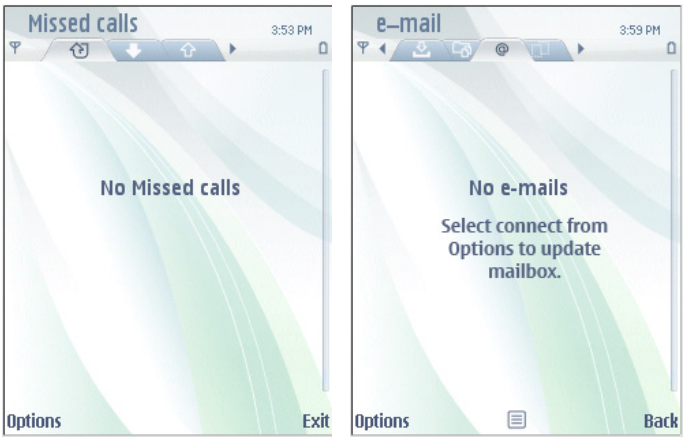


Figure 33 Empty list: the primary font used for informing the user of an empty list (left), the secondary font (right) used in addition to provide the user with further information (used only in cases when genuinely useful additional information is available)

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

8.1.3 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 required amount. 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 when moving with navigation keys. 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 for the normal keypad tone.

8.1.4 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. The 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.

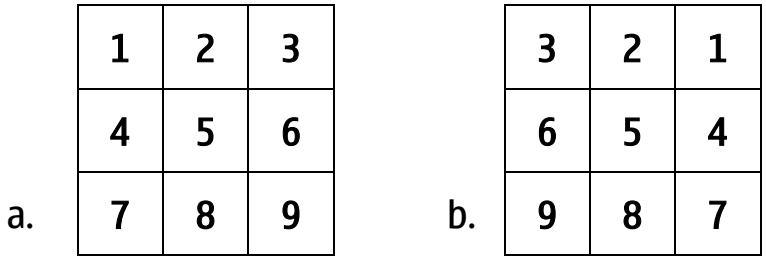


Figure 34Default grid-filling order:
a) Left-to-right UI languages
b) Right-to-left UI languages

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.

  Note: 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 focus. If all cells in the row are empty, the search continues in 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 in the same row. If there are no filled cells in that direction in 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 in 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 vice 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.

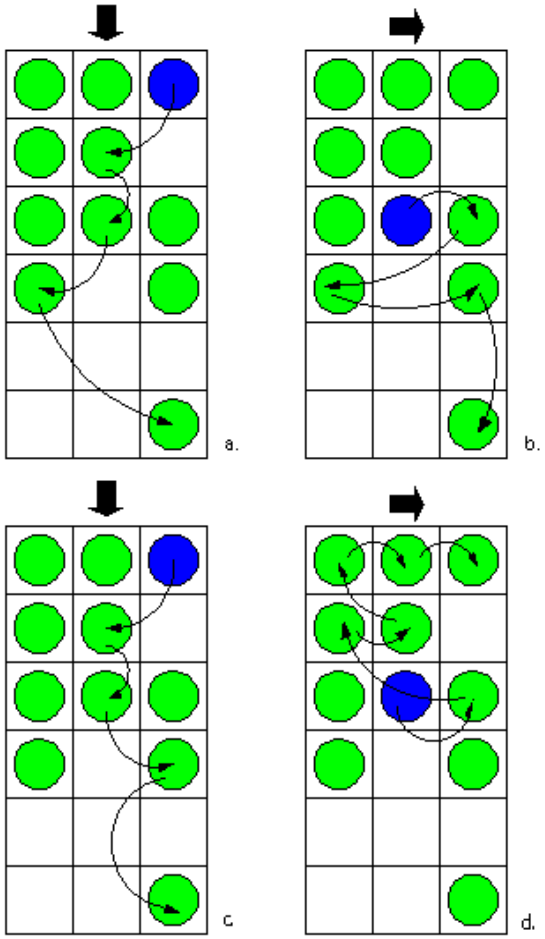


Figure 35 Examples of moving the focus in a grid
a: Left-to-right UI language; only scroll down commands used
b: Left-to-right UI language; only scroll right commands used
c: Right-to-left UI language; only scroll down used
d: Right-to-left UI language; only scroll right used

8.2 List types

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

- Menu list
- Selection list
- Markable list
- Multi-selection list
- Setting list
- Hierarchical list
- Hierarchical column list
- Form

Lists belonging to one category may have different graphical appearances. See section List layouts for information about different looking list items.

On the whole, list items and commands should not be mixed, but in cases where a 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.

8.2.1 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 are the following:

Table 16 Default key events in a menu list

Key	Action
Arrow up / down	Moves the focus in the list.
Arrow left / right	Ignored (unless there is a submenu; see section Options menu).
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.

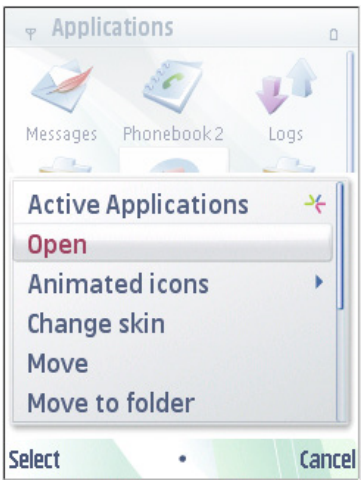


Figure 36 Options menu

8.2.2 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 Options menu).

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

- Select the item in focus. This should happen whenever it is assumed that it is clear to the user 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.
- Open the context-sensitive 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-sensitive 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 37 Select action opens a view

Figure 38 Select action performs a command

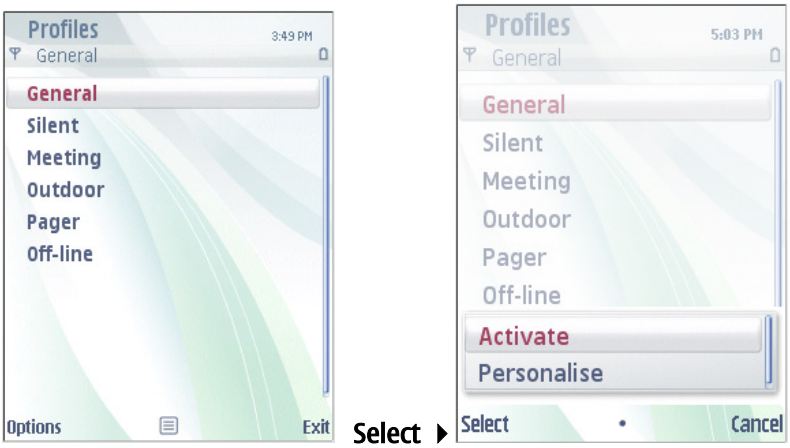


Figure 39 Context-sensitive Options menu opened with the Selection key

The keypad functions for selection lists are as follows:

Table 17 Default key events in selection list

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 Interaction with other hardware components).
Selection key	Selects the item; see the discussion above.
Left softkey (**Options**)	Opens the Options menu (see section Options menu).
Right softkey (**Back**; **Exit**)	Backstepping (see section Interaction with other hardware components).
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 Markable list).
Numeric keypad	May be ignored, or may have specific functions within the state.
Other keys	Do the default action of the key.

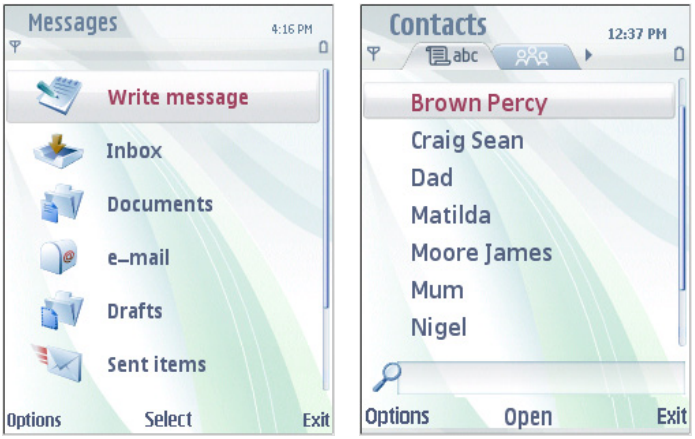


Figure 40 Selection lists

8.2.3 Markable list

A markable list is a selection list with an 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 perform certain operations more quickly and with less key presses than by performing the command separately for each of the items.

Marked items are indicated graphically.

Figure 41 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 Hash (#) key pressed while using the Arrow keys and the **Selection** key in the following way:

- Pressing the Hash (#) key marks the current item. This is a toggling function, so pressing the Hash (#) key on a marked item unmarks the item.
- Pressing an Arrow key (up or down) while holding down the Hash (#) key marks both the current item and the one onto which the focus moves. If the user keeps holding the Hash (#) 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 Hash (#) key down.
- Items can be unmarked by pressing the Hash (#) key together with scrolling: if the user starts holding the Hash (#) key down when on a marked item and then scrolls, all the scrolled items become unmarked.
- Several marking actions can be done subsequently. The user can mark an item, then release the Hash (#) 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 backstepping. 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 the Selection key when there are marked items on the list opens the context-sensitive Options menu containing only those functions that apply to multiple items.

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

Devices that have the optional Edit key perform these same functions by using the Edit key and arrow keys instead of the Hash (#) key.

The user may access the Options menu to perform functions on all marked items at once. When the user has one or more items marked, the Options menu does not contain items that apply to one item only (such as **Open**). Exceptions to this are **Help** and **Exit**, which should appear in every Options menu. 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 executed, 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 also includes **Mark all** and **Unmark all** options.

For other keypad actions, see section Selection list.

8.2.4 Multi-selection list

Multi-selection lists are used when the purpose is to emphasise that it is possible to select several items from a list at the same time. Typically, there is an on-going operation 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 with 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.

a) 

b) 

Figure 42 Multi-selection lists: in the Main pane (a) and in a setting editor (b)

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

Table 18 Default key events in multi-selection list

Key	Action

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.

8.2.5 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 43 Setting list

A setting item can be adjusted by selecting it in the same way as selection list items are selected in general; that is, by pressing the Selection key. 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.

Table 19 Setting item types

Setting	Description
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 this 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.

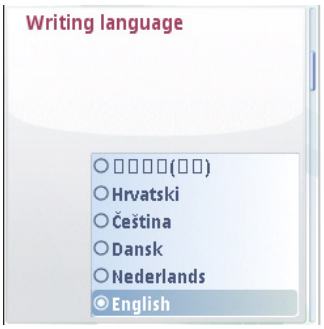


Figure 44 Pop-up setting

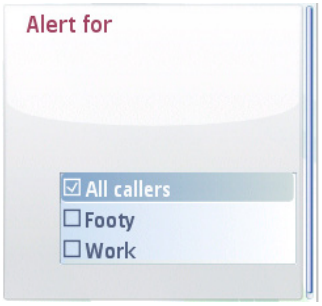


Figure 45 Multi-selection setting

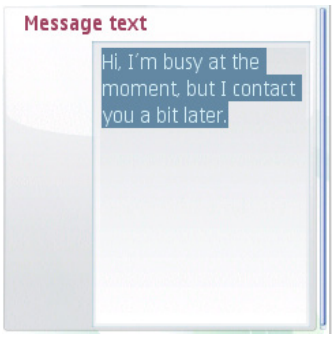


Figure 46 Text setting

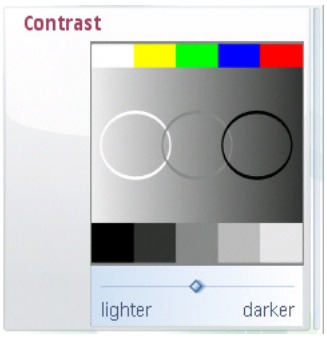


Figure 47 Slider setting

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

Table 20 Default key events in setting editor

Key	Action
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 the slider value (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, it contains the editing indicators.

Further guidelines:

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

- 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 Control panel and settings.)
- 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) must 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.

8.2.6 Hierarchical list

The hierarchical list component (a 'tree list') is available for use in dialogs that deal with moving items from one folder to another (or copying them). The main use situation for the component is File Manager folder and file browsing, where the actual data is being showed in the same view. The hierarchical list component allows users to browse through their folder content by having a list and its content available simultaneously: the user can simply expand a listed item (folder) to show its content.

Unlike the other S^3 list types, the hierarchical list component is not structured on the A – D column structure. Visually, the hierarchical levels are connected with vertical lines (see Figure 48), and the folders in each hierarchy level are shown before the files on that same level. Scrollbars function as usual (vertical bar is always present, horizontal bar is available when necessary), and the list does not loop by default, although it is possible to enable looping. Marquee scrolling is supported, but if not used, the names of folders and files are truncated where necessary. The height and the vertical distribution of the folder/file texts on the list is the same as in normal lists. Hierarchical lists can be used both in the Main pane and as pop-ups.

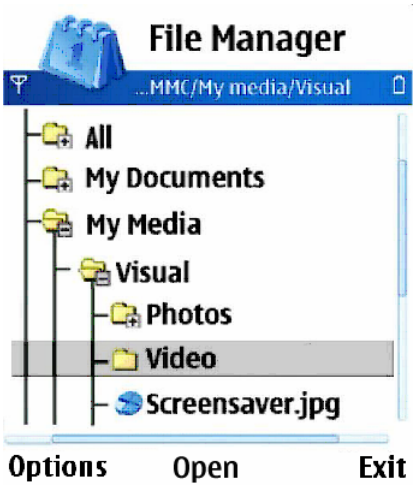


Figure 48 Hierarchical list component

Due to the need for horizontal scrolling, it is not possible to use tabs in the Navi pane. Typically the hierarchical list component is capable of showing three hierarchy levels in portrait orientation, but this may vary according to the display sizes and resolutions as well as the display orientation. The order of folders (sorting order) on each level is defined by the application.

General navigation:

- Arrow keys either expand collapsed folders (Arrow right: the focus does not move yet), collapse expanded folders (Arrow left: one press to move the focus onto the parent folder when there is one, second press to collapse the contents), or move the focus into the direction of scrolling.

- The Selection key expands collapsed (or vice versa) folders or opens files under focus.
- Commands for expanding, collapsing, and opening folders or files are also available in the Options menu.

8.2.7 Hierarchical column list

The hierarchical column list is used for browsing in one view files and folders that are on different levels in the file system. This enables, for example, showing subtitles among the list items, and collapsing and expanding data items under it.

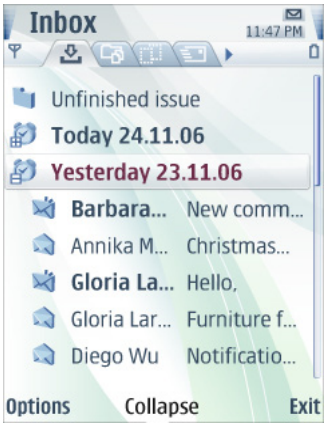


Figure 49 Hierarchical column list component, where the subtitle 'Yesterday' is expanded

The hierarchical column list can have three different row types:

- Subtitle row
- Core data row
- Simple data row

A subtitle can have subitems of only one type, thus both core data and simple data items cannot exist under the same subtitle. It is recommended that only one type of subitems would be used in the same view.

To enable more information in a single row, the core data row can be divided into two core data columns. The most important information should be shown in the first column due to more space, for example, showing email sender information in the first column, and subject text in the second column. In landscape, there can also be a third column in use.

General navigation:

Expanding and collapsing can be done in subtitle rows:

- Arrow key: Tabs can be used together with a hierarchical column list, and thus arrow keys are used for moving between tabs. If there are no tabs, the arrow keys expand a subtitle item (Arrow right, focus is not moved) and collapse an expanded subtitle item (Arrow left, focus is not moved).
- The Selection key expands the collapsed (or vice versa) subtitle item in focus.
- Commands for expanding, collapsing, and opening folders or files are also available in the Options menu.

8.2.8 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.

8.2.8.1 Always-editable forms

In case the **View** state is not useful, the form can be specified as **Edit** state-only. This way 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.

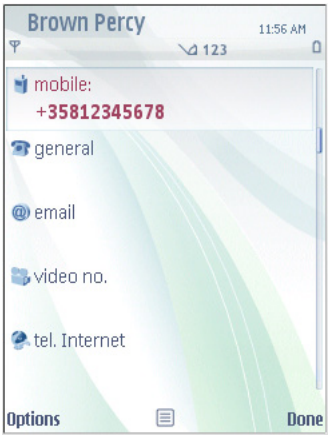


Figure 50 Form in the **Edit** state

8.2.8.2 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.

8.2.8.3 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 only in the **Edit** state. 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 List layouts 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 as 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:

Table 21 Default key events in editing state of a form

Key	Action
Arrow up / down	Moves the focus between form items (when in a text field, moves the cursor within an item line by line).
Arrow left / right	• In a pop-up field: changes the value without opening the list. • In a text field: moves the cursor character by character. • In a slider: adjusts the slider value.
Selection key	• In a pop-up field: opens the list. • In a text field: opens the context-sensitive 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 table:

Table 22 Form field types

Field	Description

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 line 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, which 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 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 51 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 51 Pop-up field

8.3 List layouts

List items can generally 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 Forms).
- The column structure of all items in a list must be similar. It is not possible to combine, for instance, 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 Column structure of lists for more information on 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.

In landscape layout, all the two-row lists can be stretched to be shown in one row. This enables showing more items in the list. In list stretching, the second row is moved next to the first row. List stretching is on by default but an application can decide to disable it.

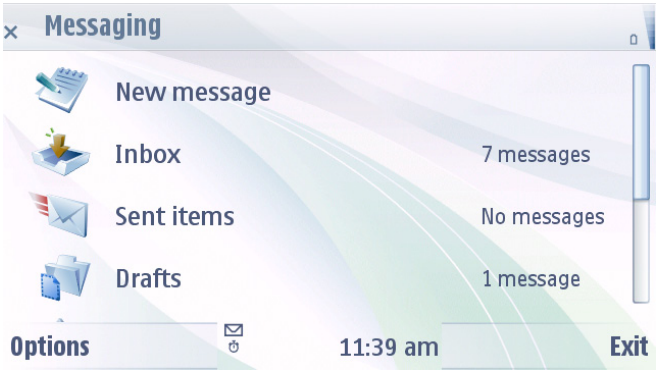


Figure 52 Two-row list in stretched mode in landscape layout

8.3.1 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 into 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. Column D is basically the very end of column C where additional indicators can be displayed (see Figure 53). The area for these indicators is not strictly an individual column in the same sense as the other three, because it can be used dynamically, item by item. All items within a list must use the same column layout.

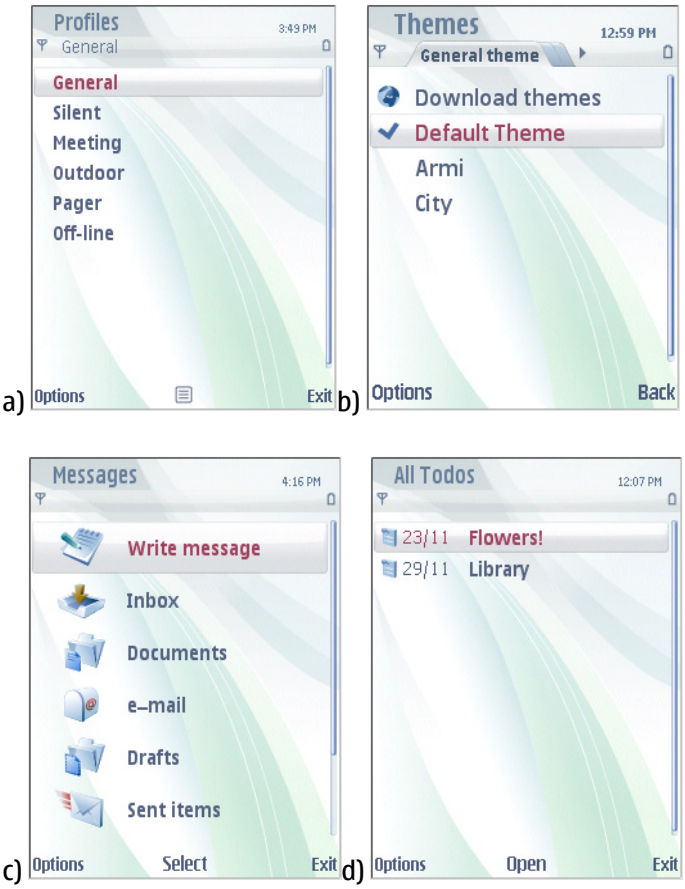


Figure 53 Different column arrangements:
a) All columns used for a text item
b) Column A contains an icon, columns B&C combined for text

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

Table 23 Standard elements within a list item associated with columns

Column	Description
Column A	• Small graphic (icon): an item property indication. • The item number (see section Numbered items).
Column B	Heading (the title or attribute of the item).
Column AB	• Heading (the title or attribute of the item). • Large graphic (for example, an icon or an image thumbnail).
Column C/BC/ABC	The main text of the item.

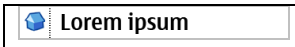
Note: The list layout can be mirrored for localisations such as Arabic and Hebrew (for example, column A is in the right edge of the pane). See section Layout changes in bi-directional languages.

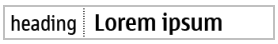
8.3.2 List item types

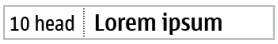
The appearance of list items can be chosen from the following types. The examples are from lists displayed in the Main pane; for most item types a similar component for use in pop-up windows is possible.

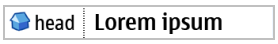
Table 24 List item types

Item	Description		
Single-line item	ABC	Main text	Primary font
[list_single_pane]	D	Item status graphics	Single color
Lorem ipsum	• Menu lists • Selection lists • Markable lists **Note:** Item status graphics in column D may be empty; they indicate, for instance, temporary item states.		
Single-line item with a number	A	Ordinal number	Primary font small
[list_single_number_pane]	BC	Main text	Primary font
	D	Item status graphics	Single color
10 Lorem ipsum	• Menu lists • Selection lists • Markable lists **Note:** Numbered lists should only be used when numbering offers some added value.		
Single-line item with a graphic	A	Item property graphic	Color
[list_single_graphic_pane]	BC	Main text	Primary font
	D	Item status graphics	Single color

<table>
<tr><td>  </td><td colspan="3">Menu lists
Selection lists
Multi-selection lists
Markable lists</td></tr>
<tr><td rowspan="3">Single-line item with a heading
[list_single_heading_pane]

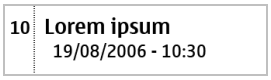
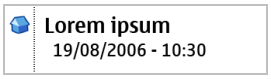
  </td><td>AB</td><td>Heading</td><td>Secondary font</td></tr>
<tr><td>C</td><td>Main text</td><td>Primary font</td></tr>
<tr><td>D</td><td>Item status graphics</td><td>Single color</td></tr>
<tr><td colspan="3">Menu lists
Selection lists
Markable lists</td></tr>
<tr><td rowspan="4">Single-line item with a number and a heading
[list_single_number_heading_pane]

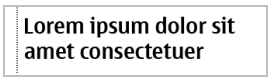
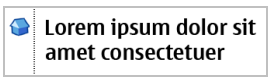
  </td><td>A</td><td>Ordinal number</td><td>Secondary font</td></tr>
<tr><td>B</td><td>Heading</td><td>Secondary font</td></tr>
<tr><td>C</td><td>Main text</td><td>Primary font</td></tr>
<tr><td>D</td><td>Item status graphics</td><td>Single color</td></tr>
<tr><td colspan="3">Menu lists
Selection lists
Markable lists</td></tr>
<tr><td rowspan="4">Single-line item with a graphic and a heading
[list_single_graphic_heading_pane]

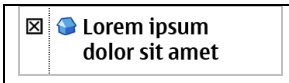
  </td><td>A</td><td>Item property graphic</td><td>Color</td></tr>
<tr><td>B</td><td>Heading</td><td>Secondary font</td></tr>
<tr><td>C</td><td>Main text</td><td>Primary font</td></tr>
<tr><td>D</td><td>Item status graphics</td><td>Single color</td></tr>
<tr><td colspan="3">Menu lists
Selection lists
Multi-selection lists
Markable lists</td></tr>
<tr><td rowspan="4">Single-line item with a large graphic
[list_single_large_graphic_pane]

  </td><td>AB</td><td>Large graphic</td><td>Color</td></tr>
<tr><td>C</td><td>Main text</td><td>Primary font</td></tr>
<tr><td>D</td><td>Item status graphics</td><td>Single color</td></tr>
<tr><td colspan="3">Menu lists
Selection lists
Markable lists
Note: A large graphic may, for instance, be an image or indicate item property.</td></tr>
<tr><td rowspan="4">Two-line item
[list_double_pane]

  </td><td>ABC (line 1)</td><td>Main text</td><td>Primary font</td></tr>
<tr><td>ABC (line 2)</td><td>Additional text</td><td>Secondary font</td></tr>
<tr><td>D (line1)</td><td>Item status graphics</td><td>Single color</td></tr>
<tr><td colspan="3">Menu lists
Selection lists
Markable lists
Note: The second line may be empty.</td></tr>
</table>

Two-line item with a number	A	Ordinal number	Primary font small
[list_double_number_pane]	BC (line 1)	Main text	Primary font
	BC (line 2)	Additional text	Secondary font
	D (line1)	Item status graphics	Single color
	• Menu lists • Selection lists • Markable lists		
Two-line item with a graphic	A	Item property graphic	Color
[list_double_graphic_pane]	BC (line 1)	Main text	Primary font
	BC (line 2)	Additional text	Secondary font
	D	Item status graphics	Single color
	• Menu lists • Selection lists • Markable lists		
Two-line item with a heading	ABC (line1)	Heading	Secondary font
[list_double_heading_pane]	ABC (line 2)	Main text	Primary font
	D (line 1)	Item status graphics	Single color
	• Menu lists • Selection lists • Markable lists		
Two-line item with a large graphic	AB	Large graphic	Color
[list_double_large_graphic_pane]	C (line 1)	Main text	Primary font
	C (line 2)	Additional text	Secondary font
	D (line 1)	Item status graphics	
	• Menu lists • Selection lists • Markable lists		
Two-line item with a graphic and a heading	A	Item property graphic	Color
[list_double_graphic_heading_pane]	BC (line 1)	Heading	Secondary font
	BC (line 2)	Main text	Primary font
	D (line 1)	Two additional graphics possible	Single color
	• Menu lists • Selection lists • Markable lists		
Two-line item with time	AB	Time	Digital font
[list_double_time_pane]	C (line 1)	Main text	Primary font
	C (line 2)	Additional text	Secondary font
	D (line 1)	Item status graphics	Single color
	• Clock alarm lists		
Two-line item, style 2	ABC (line 1)	Main text	Primary font
[list_double2_pane]	ABC (line 2)	Main text continues	Primary font

	D (line1)	Item status graphics	Single color
	• Menu lists • Selection lists • Markable lists **Note:** One text item that can extend to a second line.		
Two-line item with a graphic, style 2	A	Item property graphic	Color
[list_double2_graphic_pane]	BC (line 1)	Main text	Primary font
	BC (line 2)	Main text (continued)	Primary font
	D	Item status graphics	Single color
	• Menu lists • Selection lists • Markable lists		
Two-line item with a large graphic, style 2	AB	Large graphic	Color
[list_double2_large_graphic_pane]	C (line 1)	Main text	Primary font
	C (line 2)	Main text (continued)	Primary font
	D	Item status graphics	Single color
	• Menu lists • Selection lists • Markable lists		
Settings item	ABC (line 1)	Setting item text	Primary font
[list_setting_pane]	ABC (line 2)	Value item status	Primary font small
	D (line1)	Item status graphics	Single color
	• Setting lists **Note:** The value item may be omitted in order to create a regular selection item within a setting list.		
Setting item with a number	A	Ordinal number	Primary font small
[list_setting_number_pane]	BC (line 1)	Setting item text	Primary font
	BC (line 2)	Value item status	Primary font small
	D (line1)	Item status graphics	Single color
	• Setting lists **Note:** The value item may be omitted in order to create a regular selection item within a setting list.		
Single-line item with two graphics	A	Item selection graphic	Single color
[list_single_2graphic_pane]	B	Item property graphic	Color
	C	Main text	Primary font
	D	Item status graphics	Single color

<table>
<tr><td> 
</td><td colspan="3">Menu lists
Selection lists
Multi-selection lists
Markable lists</td></tr>
<tr><td rowspan="5">Two-line item with a graphic and a large graphic, style 2
[list_double_graphic_large_graphic_pane]

 
</td><td>A</td><td>Item selection graphic</td><td>Single color</td></tr>
<tr><td>B</td><td>Large graphic</td><td>Color</td></tr>
<tr><td>C</td><td>Main text</td><td>Primary font</td></tr>
<tr><td>D</td><td>Item status graphics</td><td>Single color</td></tr>
<tr><td colspan="3">Menu lists
Selection lists
Multi-selection lists
Markable lists</td></tr>
</table>

There is a corresponding pop-up window list item for the following components:

- single-line item
- single-line item with a number
- single-line item with a graphic
- single-line item with a heading
- single-line item with a number and a heading
- single-line item with a graphic and a heading
- single-line item with a large graphic
- single-line item with two graphics
- two-line item
- two-line item with a number
- two-line item with a graphic
- two-line item with a heading
- two-line item with a large graphic
- two-line item, style 2
- two-line item with a graphic, style 2
- two line item with a large graphic, style 2

8.3.3 Numbered items

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

Figure 54 Numbered list item

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

8.3.4 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 of this 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.

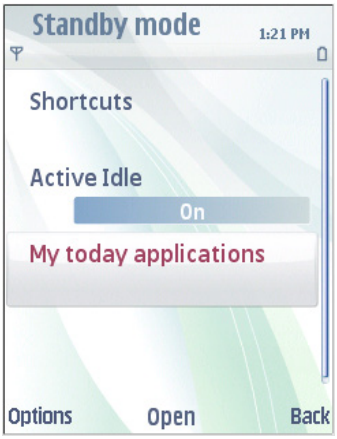


Figure 55 Setting list containing a non-setting item (My today applications) to access another view

8.4 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 might be possible with lists.
- Grid layouts are not as standardised 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:

Table 25 Grid types

Grid	Description
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.

8.5 Find pane

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

Figure 56 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 at the end of the string in the Find pane.
- The user can only add and remove characters at 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.

8.6 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.

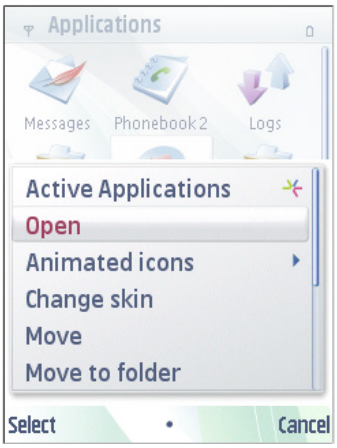


Figure 57 Options menu

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 Menu list).

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, except for the Control pane.

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

8.6.1 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.

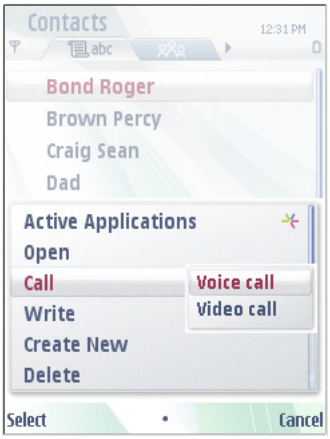


Figure 58 Submenu in the Options menu

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; a submenu cannot contain another submenu.

8.6.2 Submenu indicators

A submenu is at times used for changing the mode or temporary settings in applications. 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.

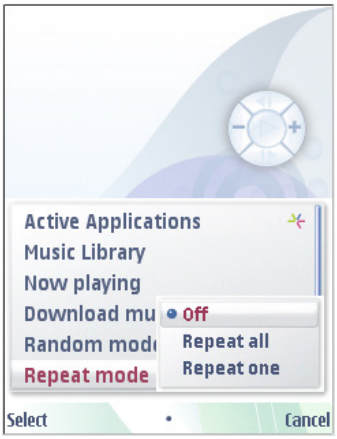


Figure 59 Submenu indicators: radio buttons

8.6.3 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 an error message must be given when the user tries to access a function that cannot be accomplished. The S^3 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.
  - One example of this are Digital Rights Management (DRM) protected files: for instance, sending commands are visible in the user interface even though sending such a file is not 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 the end-user's learning of the DRM concept.

8.6.4 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:

Table 26 Other Options menus

Menu	Description
Context-sensitive Options menu	The Selection key opens this context-sensitive Options menu when there is no single intuitive function (such as opening the item in focus) for it. The context-sensitive 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-sensitive 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-sensitive Options menu could be **Send** and **Add recipient**. In case there are marked items in a list, the context-sensitive 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-sensitive 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 Editing menu.

8.6.5 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-sensitive 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 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.

Table 27 Options menu item order

Menu item		**Description**
Active applications		Access to task swapper.
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.
Send now		Opens a messaging editor when an address for immediate sending is available, or is used for immediate sending when such data is available (for example, in the message editor).
Send ▸		Open a **Send** list query that contains the following **Send** options.
	Message	Send object via message.
	Audio message	Send object via audio message.
	E-mail	Send object via email.
	SyncML mail	Send object via SyncML mail.
	Postcard	Send object via Postcard.

	MMS Upload Services	Service provider item for uploading. All MMS upload services are displayed consecutively before local connectivity items.
	Bluetooth	Send object via Bluetooth.
	IR	Send object via Infrared.
	Other installed MTMs	Send objects via some other available MTM.
+ + +		
Create new		Initiates the creation of a new item (for example, a message in the Messaging application or a calendar event in Calendar). When more than one type of item can be created, a submenu may be used to select the type.
Edit item		Enables editing of the current item, for example, a form, or an individual item in a list.
Edit text ▸		Opens a submenu for text editing options.
	Copy text	Start text copying (copy mode).
	Cut text	Start text cutting (copy mode).
	Copy	Copy the selected (highlighted) text.
	Cut	Cut the selected (highlighted) text.
	Paste	Paste text.
Delete		• 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. The item is placed 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 list editing options.
	Copy item data	Copy data from a list item (for example, copy a phone number from Phonebook's view onto the Clipboard).
	Mark / Unmark	Mark or unmark the current item, depending on the current state.
	Mark all	Mark all items in a list.
	Unmark all	Remove marks from all marked items in a list.
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	Find a telephone number among text.

	Email address	Find an email address among text.
	URL	Find a URL among text.
+ + +		
View info (2)		View detailed information about the current item. The item is placed in this location in applications where this is a low-priority function.
Add to Favourites		For adding a link to the current application or a document to the pinboard.
Text size ▸		Change the text size in application-specified views (see section Display for more information on content zooming). Opens a submenu with an indicator.
	Automatic (default value)	Uses the text size setting value currently set in General Settings.
	Large	View text in large size. Overrides the value set in General Settings.
	Normal	View text in normal size. Overrides the value set in General Settings.
	Small	View text in small size. Overrides the value set in General Settings.
Settings		Access to application settings or context-dependent settings.
Advanced		A submenu containing rarely used items.
Extra functions		For displaying extra functions provided by application interworking. If more than one extra function is available, they are displayed in a submenu.
Writing language		Changes the input language in editors.
Help		Opens Context-Sensitive Help.
Exit		Mandatory. Exits the application.

8.7 Toolbar

The toolbar offers quick access to some of the key functions in the application view. All of the functions must also be found from the Options menu.

A tooltip, an information pop-up containing information regarding each button, must be shown.

The following toolbar button types can be used:

- **Command button** offering direct action (functions/view access) like **Send** or **New contact** with a button press feedback. A related tooltip indicates the function of the button. A command button may have a long press functionality included, such as **Fast forward**.
- **On/Off button** can be used for active/inactive type of functions such as **Loudspeaker** or **Font italics**. The button has a related latched-down effect. A button icon graphic and a tooltip indicate the function of the button, and they stay the same regardless of the button state. No long tap allowed.
- **Mode toggle button** can be used for switching between several modes as in *Flash*. The button icon graphic and tooltip change, and should indicate the current status.

Toolbar buttons are view-specific, thus within one application some views may have a toolbar and some not. It is recommended, though, to keep the toolbars as consistent as possible within an application. Changing toolbar buttons within one view is not allowed, but a button can be dimmed in case a function is not available, for instance, depending on the focused item in the view or currently available services. An application may change the buttons according to the product concept.

The order of the toolbar buttons in different orientations is according to the reading direction, so that the first item is placed topmost when the toolbar is vertical and leftmost when the toolbar is

horizontal in Western languages. The appearance of functions in the toolbar should follow the order presented below.

Table 28 Appearance order of toolbar functions

Function type	Description
Reply	Replying to a message.
Send	For sending the content .
Create new	For creating a new message, note, folder, contact, or some other item.
View switching	For functions such as Go-to or View mode switching.
Editing	For switching to editing mode.
Opening extension	For opening the toolbar extension in case all the functions within the extension are a collection of options of similar type and can be grouped under one command, for instance, the Insert extension for inserting an image, video, a presentation, and so on. For more information about extensions, refer to sections Floating toolbar extension and Toolbar extension in touch UI.
Delete	Deleting the item in Main pane.
More -extension	For opening the toolbar extension 'More' in case there are various kinds of functions within the extension that cannot be grouped.

Application-specific commands that do not fit into any function type described above are to be placed between Create new and Opening the extension in the order list.

The application may decide the initial state of the On/Off and Mode toggle button in application launch. They can also be reset or resumed from the previous session.

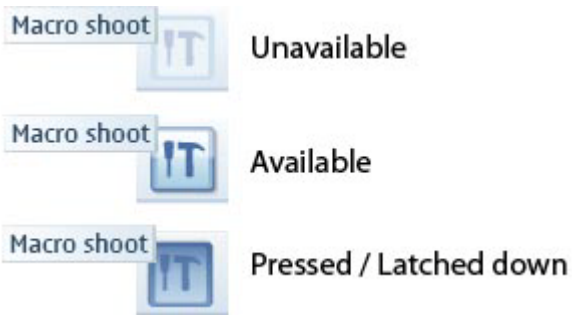


Figure 60 Toolbar button states: example of On/Off type of button

8.7.1 Floating toolbar

A floating toolbar is a component for showing the key functions on top of the Main pane or full screen content. It can be used in a non-touch UI, but also in touch UI. For more information about touch-specific issues related to the floating toolbar, refer to section Toolbar component in touch UI.

A floating toolbar owns the focus, thus it can only be used if other content in the view does not need focus. For instance, a floating toolbar cannot be used with lists and grids.

Opening and closing the floating toolbar can be done with the **Show toolbar** / **Hide toolbar** command via the Options menu. In this case, the toolbar is open by default.

A floating toolbar can also be designed to be opened via MSK. For closing the toolbar, there must then be a **Hide** button or the toolbar is closed after selecting a function. Navigation keys can be used as a

shortcut for opening and closing the toolbar, when not used for anything else in the view. Timeout can be used for closing the toolbar.

There can be 3 to 6 buttons in the floating toolbar, depending on the resolution used. Thus, the most important functions must be designed to be the first ones in the toolbar, so that they are visible in case only three buttons fit the layout, for instance.

When the toolbar is open, it owns the Selection key for selecting the focused toolbar button. The toolbar does not own the right and left softkeys. The user can move the focus over the toolbar buttons with the Arrow keys (left and right). The focus loops, and the toolbar itself does not scroll.

8.7.1.1 Floating toolbar extension

The toolbar button can be reserved for opening a toolbar extension, where more options can be found. The placement of the extension button within the toolbar can be found in the appearance order in Table 28. The button has a default graphic, which the application can change if needed.

Toolbar extension buttons are for similar actions as buttons in the toolbar, so the same types and rules apply for toolbar extension buttons. An extension should be used for accessing a set of logically similar options, for instance, when selecting content to be inserted in a message, or for adjusting imaging session-related values in the camera.

An extension can have 2 to 12 buttons (3x4 in landscape and 4x3 in portrait). The filling order follows the grid filling rules, thus the most important one should be placed first. In a floating toolbar extension, the last extension button is reserved for closing the toolbar.

8.8 Preview pop-up

Preview pop-up is a floating component used for displaying more detailed data about an item that has focus in the Main pane.

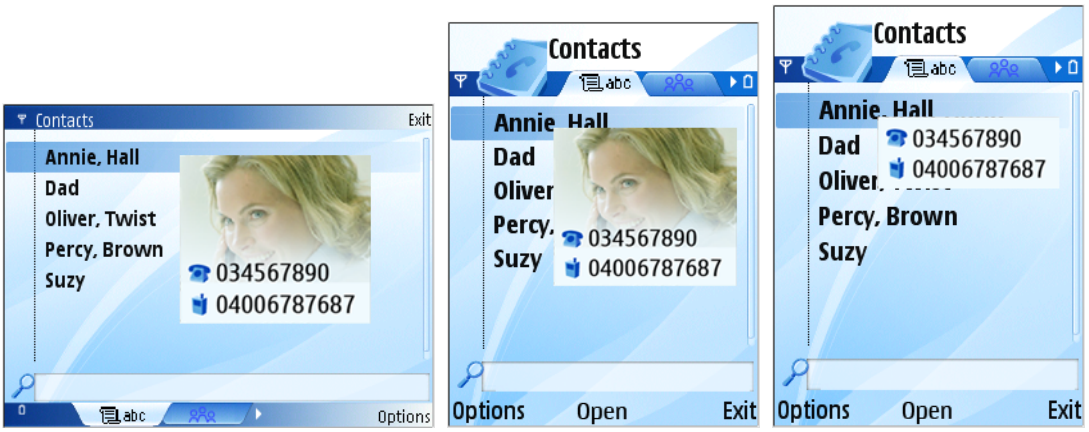


Figure 61 Floating preview

The preview is displayed with a 0.6 second delay after the focus is moved (for example, scrolling down a list with the Arrow keys) to avoid too rapid and continuous flicker. The default timeout for closing the preview pop-up is the same as for the Information pop-up, but it is possible to adjust it if necessary.

Preview pop-up contents cannot be scrolled, and the component must not contain any buttons or other elements that make it appear interactive: the component is the same in both standard and touch-enabled devices, and should the component be used in a standard non-touch device, its elements should not look touch-enabled. There can be a maximum of five text rows in a preview pop-

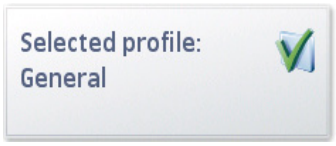
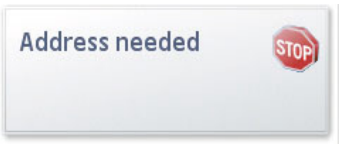
up, and it is possible to place a small icon at the start of the rows. It is even possible to have more than one icon on a row instead of text, if necessary. The number of images in a single preview pop-up is limited to one, and it can be accompanied by up to two rows of text.

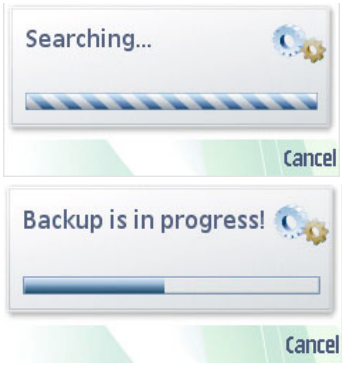
A preview pop-up always shows duplicate data in a given application to ensure that the user does not miss any content or functionality even if a given product does not utilise the preview pop-up (that is, it has been switched off). Owing to the special nature of the preview pop-up, it is important to thoroughly consider its genuine usefulness in various applications. Furthermore, the component should not be confused with the Information pop-up note, even though the data displayed in these two components can at times be identical.

8.9 Notes

A note is a feedback component that informs the user about the current situation. A note contains text and possibly a graphical element. The softkey labels are typically empty as notes do not require user input, although the user can dismiss most notes by pressing any key.

Table 29 Note types

Note	Description
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 and progress notes 	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, for instance, **Cancel** is provided.

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. 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. This way, the user must press a key to dismiss the information, giving the user time to read and think about the notification.

8.10 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 in the **Idle** state.

The user can respond to a 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.

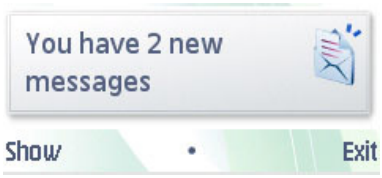


Figure 62 Soft notification indicating new messages

Soft notifications are only displayed 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. However, 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 Home / Menu key during a soft notification; in this 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 62 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 Queries).

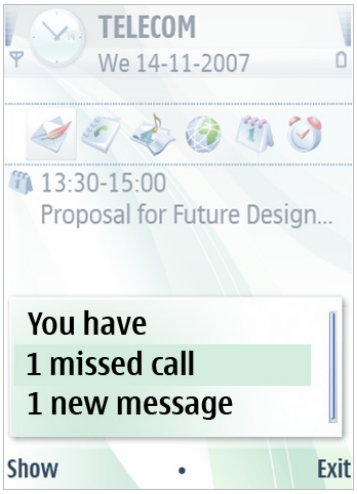


Figure 63 Grouped soft notification. The height of the window is dynamic and depends on the number of lines in the list.

8.10.1 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 from a grouped soft notification, the item becomes interpreted as obsolete, and will not reappear. If the 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 the items it contains are discarded.

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

8.10.2 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.

8.11 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 one second after the user has moved the focus onto 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.

Figure 64 Information pop-up related to the focused day

8.12 Queries

A query is a state where the software waits for user input. Queries are used in situations with more than one way 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 in a list.
- **Grid** query: a question with a selection of more than two possible (predefined) responses in a grid.
- **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.

8.12.1 Confirmation query

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

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.

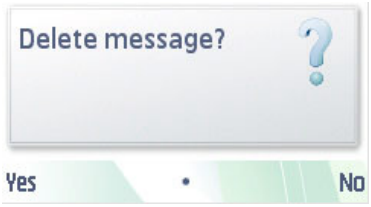


Figure 65 **Confirmation** query

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.

8.12.2 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** (hidden when moving with touch) 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 List layouts.

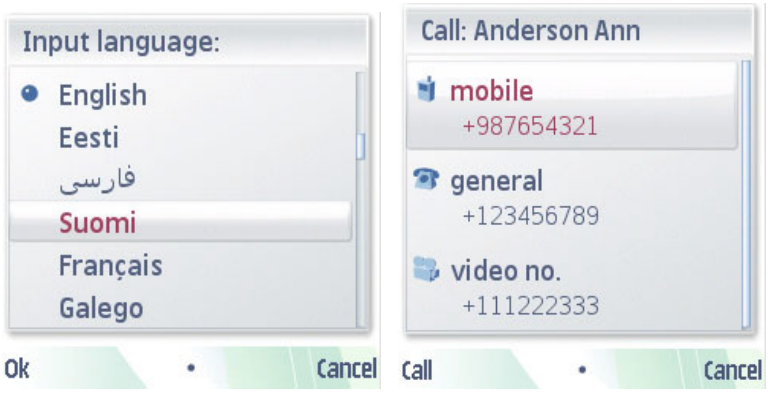


Figure 66 **List** query

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

8.12.3 Grid query

A grid can be used in a query instead of a list. The **Grid** query function is otherwise identical to a **List** query (section List query).

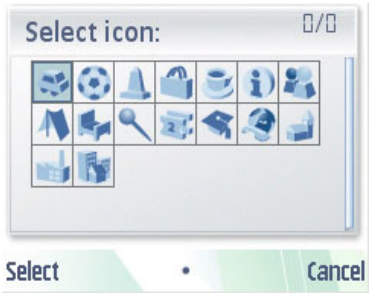


Figure 67 **Grid** query

8.12.4 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 (using the Selection key to mark/unmark the checkbox). See section Multi-selection list 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

8.12.5 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, for instance, 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 softkey 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 only for deleting characters.

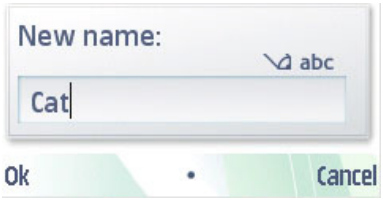


Figure 69 **Data** query

8.12.5.1 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.

8.12.5.2 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 second; 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. Note that in landscape orientation it may not be feasible to use a query with multiple fields and alternative solutions may be necessary; for example, launching one query first for user name, and on pressing **OK** another query is launched for the password.

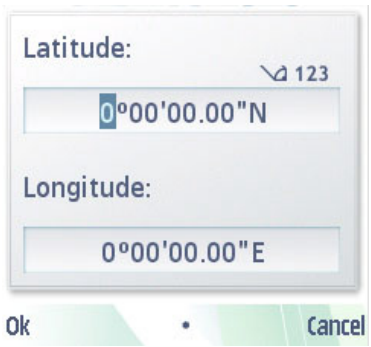


Figure 70 **Data** query with two input fields

8.12.6 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 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. Note that there should be no more than three links in a single query.

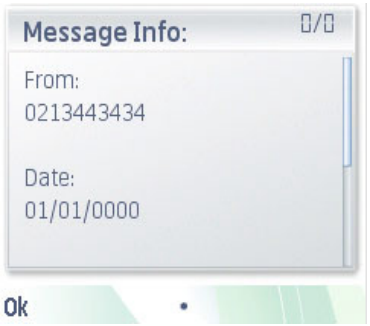


Figure 71 **Message** query

8.13 Slider pop-up

The Slider pop-up component can be used for situations where a value is changed with a slider, for example when changing zoom or volume level. The common Slider pop-up component has a set of elements from which to select and modify the slider in question, such as value ranges, whether to have the current value as text included, or additional icons related to the slider.

For an application-specific slider pop-up, the application may define ways to open the pop-up. For a common pop-up such as Volume Control, ways to open the pop-up are defined. The Slider pop-up has a default timeout after which it disappears, but for good reason this timeout can be overridden by the application.

Instead of using the pop-up, the application may define the slider to be always visible, fixed in the layout. If the slider is fixed in the layout, the pop-up is not shown.

The orientation of the slider pop-up can be either horizontal or vertical. The orientation and location are always the same across applications, unless an application uses the slider as fixed in the layout. The value growth direction in Slider is always either up or right, depending on the Slider orientation in the UI.

8.13.1 Volume Control pop-up

In case the hardware has no dedicated volume keys, the audio volume is adjusted using the Arrow up and Arrow down keys. When the user is controlling the volume, the key event brings the Volume Control pop-up to the view, and the pop-up disappears after a timeout when the user has done the adjustment.

The application may need up/down key events for controlling some other component, such as list. In this case, it needs to be decided whether that view can be left without Volume Control or whether the Volume Control component is to be accessed via the Options menu or toolbar.

In touch UI, it is possible to activate Volume Control via the Universal indicator pane. Volume is adjusted with touch by dragging the slider handle, and audio can be muted by tapping the icon in the UI component. For more information, see Touch support for common S^3 components.

The Volume Control UI component may also be always available in the view. In this case, it is fixed in the layout.

Figure 72 Volume Control UI component

8.14 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: in non-touch UI the user cannot point at an indicator to perform actions.

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

8.14.1 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. Indicators consist of a bar graph that indicates the current battery or signal level and an icon that describes a battery, or in signal indicator the connected cellular network and its connection state.

8.14.2 Universal indicators

Universal indicators are small graphical icons. They inform the user about issues such 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 universal indicators are displayed in the universal indicator area in the top-right corner (next to the Battery pane) in the Status pane (see Figure 73).

Within applications other than Phone, status indicators appear in the Universal status indicator pane, which is combined with the clock display. 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.

Figure 74 Universal status indicators displayed in the top-right corner in nHD portrait orientation

8.14.3 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:

Table 30 Navi pane indicators

Indicators	Description
Left and right arrows  Figure 75 Arrows in the Navi pane	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 end 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  Figure 76 Editing indicators in the Navi pane	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.

8.14.4 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 section Editing indicators for an example).

8.14.5 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.

8.14.6 Operator indicator

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

8.14.7 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 utilised as indicators.

9 S^3 UI with rotation

The S^3 UI can rotate between the portrait and landscape layout. This chapter describes the guidelines for designing landscape-aware application user interfaces. On the whole, the primary guideline in the S^3 platform is that no specific UI design is necessary for landscape orientation and mode. Instead, applications designed according to the guidelines described in this chapter should work in both the landscape orientation and the default orientation. This chapter does, however, describe some exceptions to this rule.

The S^3 in landscape orientation follows the bottom softkey style. In the bottom softkey style, the layout of the normal portrait UI is merely stretched horizontally to fit the screen, leaving the softkeys below the screen as in standard portrait orientation.

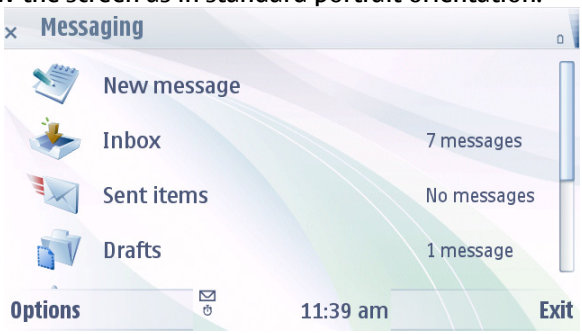


Figure 77 S^3 UI rotated to landscape in the bottom softkey style

9.1 Interaction

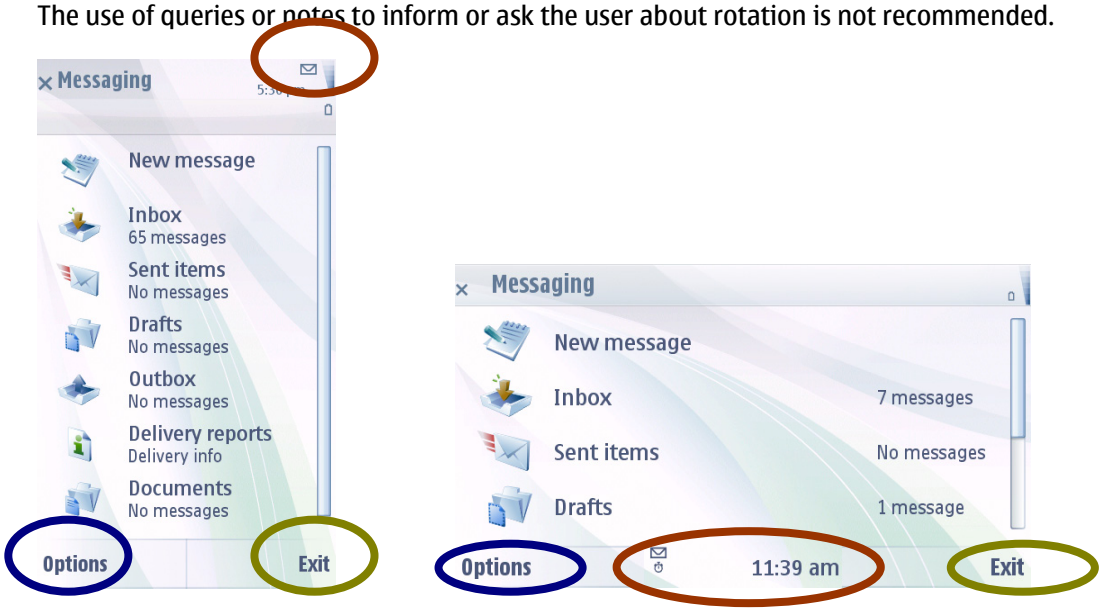
9.1.1 Rotation

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 the 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.

Applications that support only one orientation, and thus trigger an 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.

An exception to this rule are applications that are launched from the landscape application itself (for example, the **Go to gallery** option in the Camera application), or embedded within the application (for instance, 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.

The use of queries or notes to inform or ask the user about rotation is not recommended.



9.1.2 Basic interaction style

The basic interaction principles of the S^3 platform in the bottom softkey style remain similar to the legacy S^3 UI style. This applies to keypad functions, typical functions of the keys, navigation (both in terms of hierarchies and the scroll and select behavior), and so on.

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

Similarly, other visual UI controls of scroll keys rotate together with the UI. For example, this applies to the Volume Control component which is also vertical in landscape mode.

Note: In some product concepts, using the numeric keys and some of the navigation 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.

10 Applications and design examples

10.1 Home screen

Home screen is the basic state of the device. Once the user has powered the phone on, it results in the home screen state after all the startup activities have been finalised. Home screen displays information about the current state of the device, including the connection status, battery level, and network operator. See chapter UI components for detailed information about the indicators.

In addition to device state information, home screen can include mini views of the applications. There is a default set of these mini applications available in the home screen, but the user and the service provider can customise the home screen content. Mini applications can be individual applications or they can work as shortcuts to the actual applications. Mini applications may update their content dynamically, for example, an email mini application may always show the three most recent emails.

Home screen can contain multiple pages. The user can add and remove pages. Pages can also be protected so that they cannot be deleted.

Figure 78 Example layouts of the home screen in nHD portrait and landscape orientations

10.2 Menu

Menu is the menu used for accessing all the applications in the device. **Menu** can be opened by pressing the Home/Menu key. In addition to direct access to applications, there can be 'meta-applications' that collect applications related to the same area under a main folder. Applications are presented as a grid of items by default; the user can also choose to use some other customized version of the **Menu** layout. The recommended grid layout is 3x4 (nHD portrait) and 6x2 (nHD landscape), 12 icons in all. Selecting an application from the menu opens the application and closes the **Menu**. See section Application handling for detailed information about opening and closing applications.

Figure 79 **Menu** with a 3x4 grid in nHD portrait

For more details on the landscape mode, see chapter S^3 UI with rotation.

10.2.1 Shortcuts

When **Menu** 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 **Menu**'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.

10.2.2 Customising Menu

The user can adjust the order of applications, as well as create folders and move applications into folders within **Menu**. It is possible to have folders within folders in **Menu**, allowing for more hierarchically organised structures. The number of levels for folder-in-folder is not limited and the folders are user-customisable (meaning that the user can add and remove folders). These managing functions are available through Options. In addition, the layout of **Menu** may be customised.

10.3 Application handling

This section describes the handling of applications and the interactions between applications in the S^3 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 **Menu**, there is never 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 therefore see the same feature being run in several different applications at the same time.

10.3.1 Opening and closing applications

Applications are typically opened via **Menu**.

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

- Using a shortcut in the home screen state.
- 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.

The user can close applications in the following ways:

- Backstepping from the application using the right softkey.
- Using the **Exit** function from the Options menu.
- Pressing the Call termination key.
- Closing application from the Task switcher. For more information about the Task switcher, see section Multitasking.

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.

10.3.2 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 **Menu** 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 **Menu** 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.

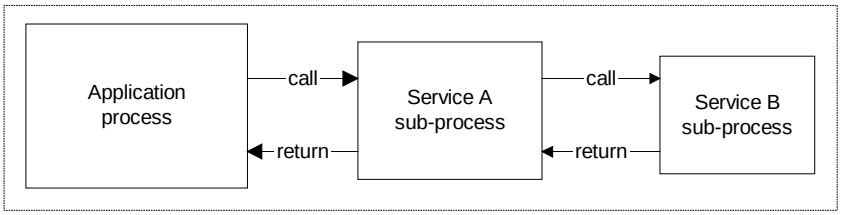


Figure 80 Nested services running within an application process. The **Back** function leads to the parent process.

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 cancelled. The user is not asked any questions; all the operations needed to bring the application to the target state are done automatically. After switching applications 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.

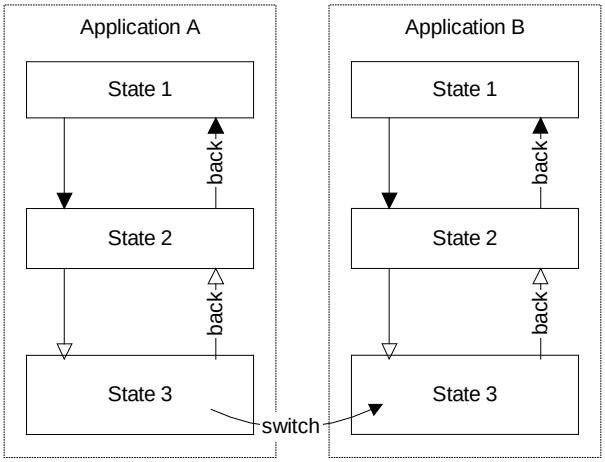


Figure 81 Switching between applications, interrupting application B. The **Back** function works inside application B, it does not lead back to application A.

10.4 Phone

Phone is the application for handling voice and video calls. It is a central application in the device. Calls can be created by dialling (or voice dialling) from the home screen state, Phonebook, or any other state in the device where a suitable number is available. Whenever there are one or more voice calls going on, the Phone application takes the place of the home screen state. Swapping between Phone and the home screen state is done via the Options menu or the task swapping window.

A contact image is shown as a background image on an incoming and ongoing call. The user can define an image for a contact in the Phonebook application.

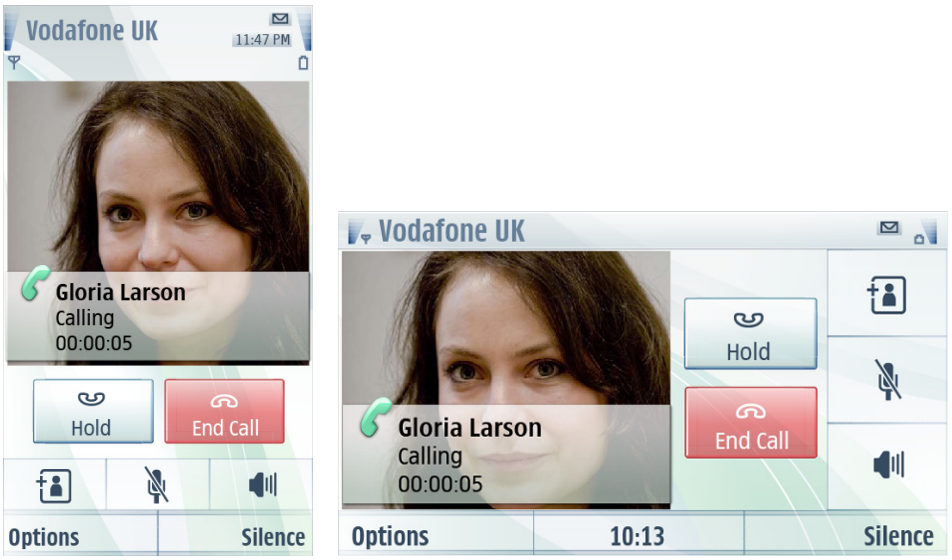


Figure 82 Ongoing call in nHD portrait and nHD landscape orientations

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

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.

10.4.1 Call handling

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

Table 31 Call handling key functions

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; activates 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 call and a held call, ends the active call and activates the held call. • A long press terminates all calls (active and held).

The basic rule is that the call handling keys accomplish the same functions described here regardless of whether the Phone application is on the screen or not, so the user does not have to 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: for example, in Messaging Centre, the Call creation key can be used to send a message). Calls can also be handled through the Options menu or as service code sequences.

10.4.2 Dialler

The Dialler application is used to make phone calls to the dialled numbers. User can open the Dialler application by pressing the number keys from the physical keyboard in home screen state. In case the device doesn't have a physical keyboard, the Dialler component will include a virtual ITU-T keypad for

dialling the number. In devices using nHD resolution, the Dialler can also be opened by selecting a Dialler shortcut from the home screen or by pressing the Send key in any state of the device, except in the home screen where pressing the Send key opens the **Dialled calls** list view from the **Logs** application and in couple of other exceptional cases.

By default, the dialled numbers are also matched to the Phonebook contacts based on the corresponding letters in the ITU-T keypad. The matched contacts are shown above the dialled numbers and the user can call the matched contact simply by selecting the contact. In case the device has a physical QWERTY keyboard, the user can type a name using the letter keys on the physical keyboard.

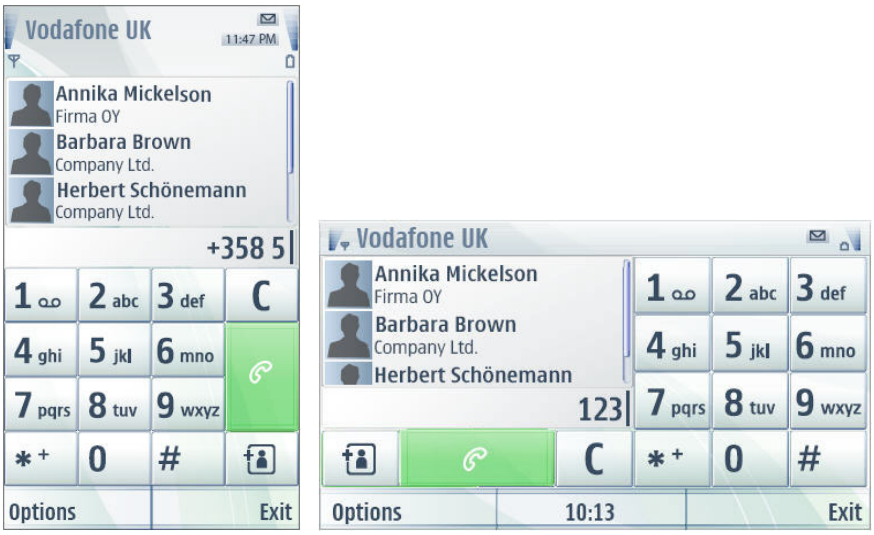


Figure 83 Dialler with virtual ITU-T keyboard

10.5 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. The data is then 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.

10.5.1 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).

10.5.2 Image viewers

A still image viewer is a tool optimised 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.

10.5.3 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 the whole image can fit in the display. 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 one scrolling step should let the user view the full image (if scrolling occurs strictly one screen at a time, an image could be split into two parts, each one in a separate screen.)
- The user should be 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.

10.5.4 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 a 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 must 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 a 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, for instance, voice memos.

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

10.6 Document handling

10.6.1 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, for instance, a number after the common name. This way it is possible to have meaningful document names without typing them for each file.

10.6.2 Saving edited data

Various kinds of documents may have different requirements concerning the behaviour of keeping or discarding the edited data. Typically, one enters or modifies data and then closes the editor, accepting the new data. However, sometimes there may be a need to do something else, for instance, 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 element. 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, the user has to select **Edit** from the Options menu. Most forms do not appear in the **View** state at all; thus the items are 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 will. The user can accept the whole form with 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 is **Cancel**. This arrangement is easy to understand.

However, there are often functions that must be accessible while editing the form, for example, adding new items to the form. In this 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 also needed, 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. Also in this case, the right softkey saves the data and returns 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. However, it is worth noting that sometimes technical restrictions, such as memory limitations, may prevent discarding all changes to the document, for example.

10.6.3 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 folders 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.

10.6.4 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 exists elsewhere. This makes it possible to access the same data from several places. A typical application for groups is a distribution list: the user 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.

10.6.5 Fetching data

It is often 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 is 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.

During a fetch operation, the left softkey is labeled **Select** and it activates the same function as the Selection key: it selects the data currently in focus. In case there is no visual focus in the UI, the left softkey is empty. 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 likely 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 name in the group, a multi-selection list is a good tool to use (see section Multi-selection list).

10.6.6 Embedded links

It is possible to initiate the downloading of content (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.

10.7 Control panel and settings

The Control panel application replaces the former General Settings application. The Control panel is not only meant for device and application settings, but it is also the place to do configuring actions to the device, for example, checking software updates. Applications that are setting or configuring type in nature are to be placed in the Control panel instead of crowding the **Menu**. For example, Connectivity applications, Device manager, and Synchronizing applications are placed in the Control panel. Control panel will also include links to all application-specific settings.

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 List types).

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:

- 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). (recommended solution)

- Use tabs to access different setting groups.

10.8 Wizards

For some tasks, it is possible and even sensible to offer a wizard for completing a complex or long task. In its simplest form, a wizard is a progression of steps that each offer two or more items to choose from in order to complete a larger task or a group of settings. The items selected are the variables needed to compile the task or setting the wizard produced. The final structure of a wizard depends to a large degree on the task or tasks it performs. Still, some common guidelines are necessary to ensure consistency over the S^3 interface:

- A good wizard shows the user an informative message at the start, explaining what the wizard does and what will be the eventual outcome from adjusting the set of variables offered in it.
- Each item or option offered to the user at the various stages of the wizard progression should be unambiguous so that there is no confusion as to the meaning of each item or how the items differ from each other.
- If possible, it is advisable to avoid steps where the user cannot make a selection but is merely informed of something that is about to take place.
- The progression logic should be simple and linear with no branching out or deep hierarchical structure: in each step of the wizard, the user simply chooses or sets an item and proceeds with the **Next** command on the left softkey.
- If there is a situation where a selection must be made before the user can proceed, the left softkey should be hidden (inactive) until the selection is made (or the user backs away using the right softkey).
- The left softkey is **Next** and the right softkey is **Back**. In other words, as the progression logic is linear, it is possible for the user to return to previous steps (and change a selection there) by using the right softkey (**Back**). The user can also interrupt the wizard without saving any of the selections at least by reversing to the beginning of the wizard.
- The user must always be informed about saving the changes and then asked (if applicable) whether to use the wizard's outcome immediately.
- Other necessary interface details:
  - In selection lists using radio buttons, none of the buttons are selected as default.
  - The page number out of the total number of wizard pages (for example, **3(7)**) should be visible in the Navi pane on each page.
  - The progression typically takes place by advancing from one common settings editor (for example, a slider editor, radio button editor, or text editor) to another.
  - Additional information on the current setting/stage can be provided in an Information note.

10.9 Help

A device with S^3 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.

11 References

[1] S60 Internationalisation and Localisation Guide	*http://forum.nokia.com/*