The User Interface of the Nokia 9110 Communicator



Issue 1
9359721
Nokia Mobile Phones
Customer Application Development group

email: 9110.sw.developer@nokia.com

TABLE OF CONTENTS

1. INTRODUCTION 1
   1.1 Document overview 1
   1.2 Glossary 1
2. GEOS SDK 2
3. NOKIA 9000/9110 DEVELOPMENT ENVIRONMENT 2
4. GENERAL FUNCTIONALITY 3
   4.1 Three pane approach 3
   4.2 Fundamental functionality rules 4
   4.3 PDA-side applications 4
   4.4 Getting started with Geos programming 5
5. USER INTERFACE 6
   5.1 General UI 6
   5.2 Indicator, Application and Command Button Areas 9
   5.3 Command buttons and special keys 9
   5.4 Notes and search functionality 13
   5.5 Graphical UI 17
6. TELEPHONE 21
7. FAX 22
   7.1 Sending faxes 22
   7.2 Some fax settings 23
   7.3 Receiving faxes 23
8. SMS 24
   8.1 SMS main view 24
   8.2 Received messages 24
   8.3 Create and send a message 25
   8.4 Concatenated short messages 25
9. INTERNET 26
   9.1 Mail 26
   9.2 WWW 30
   9.3 Telnet, Terminal and Text Web 36
   9.4 General guides for developing Internet applications 36
   9.5 Task switching in the Internet application 39
   9.6 How an application should react in different error and informative situations 39
10. CONTACTS 41
   10.1 Main view 41
   10.2 Contact card image 42
   10.3 Individual ringing tones 42
11. NOTES 43
   11.1 Notes main view 43
   11.2 Note editor 43
   11.3 Received faxes, Downloaded pictures and DOS files 43
   11.4 Send and Print functionalities 43
12. CALENDAR 45
   12.1 Month view 45
   12.2 Week view 45
   12.3 Day view 45
13. SYSTEM 47
   13.1 Security 47
   13.2 User data 47
   13.3 PC Connectivity 48
   13.4 Wireless Imaging 48
   13.5 Infrared receiving and remote synchronisation 48
   13.6 Memory issues 49
   13.7 How to make a Software inquiry program 51
14. EXTRAS 52
   14.1 Voice recorder 52

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

14.2 Taskswitching in the Extras application 52
15. HELP TEXTS 53
15.1 Help view 53
15.2 How to make help files 54
15.3 Help style guide 55
16. INDICATORS 56
16.1 Indicator application interface 57
16.2 How to set an application icon 58
16.3 Setting application text 59
16.4 Setting the number of unread received documents 59
16.5 Changing the number of unread received documents 59
16.6 Setting the number of outgoing documents 59
16.7 Changing the number of outgoing documents 59
16.8 Setting custom notes 60
16.9 Setting visibility of main window 60
17. DOCUMENT OUTBOX 61
18. LOCALIZATION 62
18.1 Resource Editor (ResEdit) Tool 62
18.2 Language versions 62
18.3 File names 62
19. APIs 65
20. APPLICATION COPY PROTECTION 68
21. HOW TO DELIVER YOUR APPLICATION 69
22. SUMMARY 70



A standard design for User Interfaces creates even happier users!

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

1 INTRODUCTION

1.1 Document overview

A user interface (UI) specification is a set of rules and conventions used to determine the interaction between a user and an application. This document is intended for developers who are writing software for the Nokia 9110 Communicator. It is recommended that developers use this user interface specification when developing third-party applications, particularly if the developer plans to get the Nokia OK logo.

The purpose of the original document "*Nokia 9000 Communicator user interface, introduction for developers,*" was to give a general overview of the Nokia 9000 Communicator UI and its functionality. The document "*Nokia 9000i Communicator user interface, new features for developers"* explains all the new modifications and new features of the 9000i over the original 9000 user interface. The document "*Nokia communicators, Software Development Guide*" combines those two documents and the developer is able to easily see the differences between the original 9000 and the 9000i versions. The document you are reading now gives a general overview of the Nokia 9110 Communicator UI and its functionality. Each one of these documents is available for developers and can be ordered by e-mail from: 9110.sw.developer@nokia.com.

If you have any questions or comments related to this document, please use the following e-mail address: 9110.sw.developer@nokia.com. This document is based on the:

- Nokia 9000 Communicator user interface, introduction for developers
- Nokia 9000i Communicator user interface, new features for developers
- Nokia communicators, Software Development Guide
- Geoworks[1] 9110 SDK CD-ROM, version 1.1
- Geoworks Nokia 9000 Communicator Programming Class Notes
- NOKIA 9000, 9000i and 9110 Communicator User manuals
- Forum Nokia http://www.forum.nokia.com/

Each document is updated separately. Nokia Mobile Phones, Customer Application Development group will update this document when needed.

1.2 Glossary

API	Application Programming Interface
CMT	Cellular Mobile Telephone
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IAP	Internet Access Point
N9kDE	Nokia 9000/9110 Development Environment Tool
NMP	Nokia Mobile Phones
PDA	Personal Digital Assistant
SDK	Software Development Kit
UI	User Interface

[1] Geoworks ® application software and GEOS ® operating system software copyright © 1990-1999 Geoworks. All rights reserved.

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

2. GEOS SDK

Nokia Communicators use the GEOS 3 operating system developed by Geoworks. The Nokia Communicator Software Development Kit (SDK) is available from Nokia. Training, support and ordering details of the Nokia Communicators SDK can be found at: http://www.forum.nokia.com.

With the Geos SDK for the Nokia 9110 Communicator it is possible to create third-party applications for the 9110. The 9110 SDK CD-ROM also includes an SDK for the Nokia 9000i Communicator. With this 9000i SDK it is possible to create applications for both the Nokia 9000i and Nokia 9000il Communicators. The existing Geos SDK versions are:

- for Nokia 9000 Communicator: versions 1.0 and 1.01
- for Nokia 9000i Communicator: version 2.0
- for Nokia 9110 Communicator: versions 1.0, 1.01 and 1.1

The SDK references in this document are for the 9110 SDK, version 1.1. The earlier versions of the 9110 SDK can be updated to 1.1 at Forum Nokia [2].

The operating system for the Geos SDK must be Windows NT. For compiling the code, you need to have the Borland C++ compiler (version 4.5 to work correctly). You also need a WWW browser for reading the SDK documents. Information about the system requirements is given in the SDK [3].

3. NOKIA 9000/9110 DEVELOPMENT ENVIRONMENT

The N9kDE tool makes it easier for new developers to enter the world of Geos programming. It provides Graphical User Interface to Geos SDK. Firstly, it offers an easy access to compilation-, debugging- and emulation tools. Secondly, it gives full source editing capabilities. Thirdly, it has direct positioning on errors.

It is important to note, particularly from a new developer´s point of view, that the Nokia 9000/9110 Development Environment speeds up application development and prototyping. It takes care of redundant programming tasks. The tool also gives an opportunity for easy creation of application frameworks.

The N9kDE offers an easier, *menu-based access* to the Geos SDK. In other words, easy access means an all-in-one environment, which includes makefile creation and compilation tools, an emulator and a SWAT debugger. There is also *window-based access* to the Geos SDK icon-importing tool. It inserts icons directly to your application and no further programming is needed.

The tool has many practical features. *Geos SDK help* has a keyword search, which provides quick access to the information you need. There is no need to be exact about what you are looking for. *The source code editor* contains syntax highlighting, smart indent, multiple undo/redo capability and line and column numbers. *The output box* with direct cursor positioning helps to locate errors in the code. In the output window the syntax highlighting makes it easy to follow the progress of compilation and other tasks. *The graphic editor* allows the addition of simple graphics to applications. It includes lines, ellipses, rectangles, round rectangles and polylines. *Packaging* is the convenient and fast way to create installation files.

The N9kDE can be downloaded at: http://www.forum.nokia.com
For technical support and further information, please contact the developer support group [4].

[2] http://www.forum.nokia.com (sdkupdate.exe)
[3] pcgeos\Techdocs\Intro\InstallSDK\index.htm
[4] 9110.sw.developer@nokia.com

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

4. GENERAL FUNCTIONALITY

The main elements are the PDA-side and CMT (Phone) -side. The user interface (UI) is accessed using the keyboard and the buttons beside the screen. The execution of different PDA-side applications using the main screen is shown in this section.

Figure 1 shows the most important parts of the user interface from a functionality point of view. The contents of the opened device are always referred to the "PDA-side" and the cover of the device is called the "CMT-side."



Figure 1. Basic elements of the user Interface.

4.1 Three pane approach

The PDA-side UI of the main screen is divided into three panes. See Figure 2.

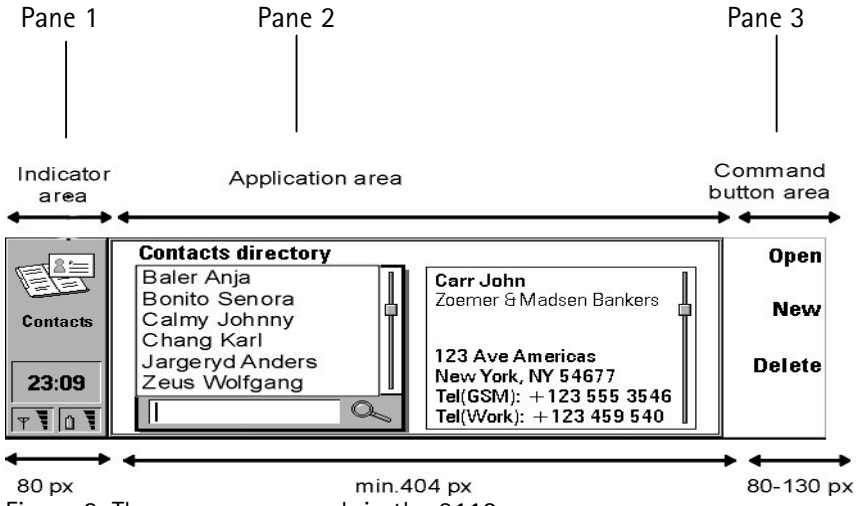


Figure 2. Three-pane approach in the 9110.

The left pane (Pane 1) is normally used to show indicators (see Chapter 16, Indicators), application symbols and application names.

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

The middle pane (Pane 2) is normally used for application views.
The right pane (Pane 3) shows currently valid commands. Four Command buttons are used to activate commands.
The size of the display is 640 * 200 pixels. The application icon and title are always centred. The maximum width of the title text on one line is 70 pixels. The title can use two lines if needed.
The command text can use two lines also. This happens only when the commands do not fit on the first line. However, the text can be divided into two lines if an extra space is needed in the application area.

MIN MAX

Figure 3. The minimum, 80 pixels, and maximum, 130 pixels, widths of the command button area. See Chapter 5.2 for more detail.

4.2 Fundamental functionality rules

There are some fundamental functionality rules:

- The user can change in any state from one main application to another by pressing one of the main application buttons. The previous main application is left in its current state when the new application is activated.
- The device always saves changes when exiting a view (if there is no Cancel button). Data are saved automatically.
- Main applications (listed in Chapter 4.3) cannot be closed. They are always running. New applications can be activated independently of the other applications.
- Please familiarise yourself thoroughly with all functions of the Nokia 9110 Communicator, including particularly the instructions for its safe operation contained in the user's guide. All software must comply strictly with Nokia safety principles (for example, flight mode to block operation of the phone interface on airplanes).

4.3 PDA-side applications

PDA-side applications usually refer to the following nine applications:

1. The **Phone** application enables call origination and receipt. It includes many advanced call-handling features.
2. The **Fax** application includes both Fax sending and receiving. Received faxes can be read, forwarded or printed. Faxes are saved as documents in the document list.
3. The **SMS** application (Short Message Service) includes both Short Message sending and receiving. Received Short Message can be read or replied to. Received and created messages are saved as documents in the document list.
4. The **Internet** application includes modem-based services, such as Mail and WWW, SMS-based services, such as Text Web, and modem-based terminal emulator connections.
5. The **Contacts** application includes contact information such as contact names, related data and general contact information. From general contact information you can find for example where and when the call was made and so on. The Contact Manager database has a key role also in the Phone, Fax, Mail and SMS applications.
6. The **Notes** application is a file manager for working with created and received messages. Notes enables file handling such as creating, deleting, opening, renaming, printing and sending.
7. The **Calendar** application includes Month, Week and Day views. It is possible to add events to special days, set alarms for them and add memos. Entries, which are not related to any specific time or day, can be added to your to-do list.
8. The **System** application includes all system-related functions, such as PC connectivity, adjustments and information about the device.

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

9. The **Extras** application is devoted to applications that do not have enough space to have their own button. Extras includes accessories, such as the Clock and Calculator. It is possible for a developer to add more accessories or games in the Extras application.
10. **Help** is context-sensitive help for the currently active view on the screen. (Help is not actually an application.)

Each of the above-listed applications is described in detail in separate sections. Phone panel functionality was ignored in the previous list. The phone panel UI and functionality are described in separate documents not listed in this document.

4.4 Getting started with Geos programming

Before writing an application, it is useful to get familiar with some fundamental Geos programming concepts. There is a tutorial section in the SDK, which provides a step by step example of how to create an application. For compiling and running an existing sample application, please see the SDK [5] and [6].

[5] pcgeos\N9110v10\Appl\SDK_9000\Hello\
[6] pcgeos\Techdocs\Intro\Tutorial\index.htm

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

5. USER INTERFACE

The purpose of this chapter is to describe the functionality of general UI elements common to many applications. There are some principles, which should be noticed when developing Third-party applications:

- pay attention to clarity and uniformity of screens
- not everything is intended to appear as to one page
- only necessary commands should be available on the screen
- there must be a Cancel button in long operations
- by following the principles and rules introduced in this document and in the SDK, you'll be able to develop "Nokia look and feel" applications (a sample application can be found in the SDK [7])

5.1 General UI

5.1.1 Name list

The name list can be found from the Tel, Fax, SMS, Mail and Contacts applications. The list shows all the entries that have been entered in Contacts and is sorted alphabetically. More information about the name list can be found in Chapter 8, Contacts. The name list is illustrated in Figure 4.

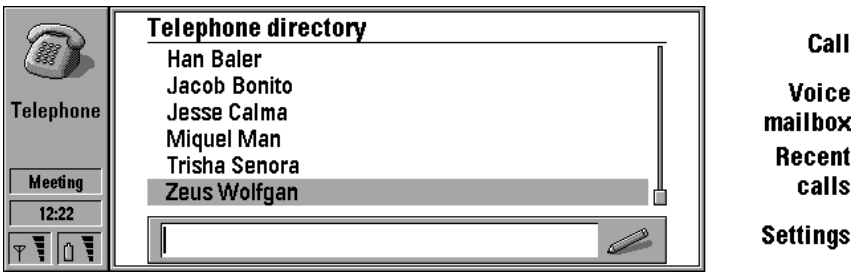


Figure 4. Name list.

5.1.2 Document list functionality

The document list component is used for displaying file directories inside the communicator. The list is divided into folders or sub-directories, which contain the files.

Folders are selected with the selection frame and then opened. An open folder includes a list of documents (see Figure 5) or "No documents" if the folder is empty.

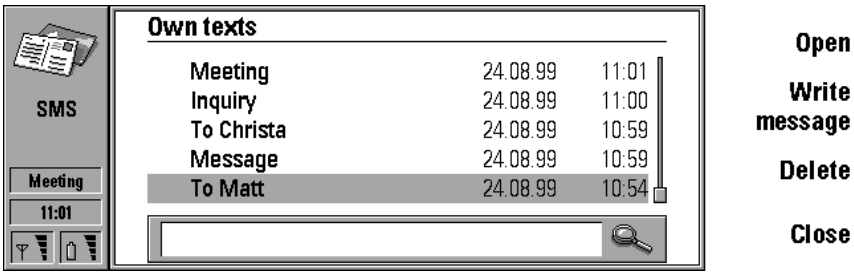


Figure 5. An open folder.

Commands related to folders vary. The document list also includes a document outbox folder. The document outbox does not include actual documents, but shows the queue of documents that are to be sent using one of the applications. The document outbox is explained in Chapter 17.

[7] pcgeos\N9110v10\Appl\Sdk_9000\Dialogs\

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

5.1.3 Document naming

New documents (created or received) are named as shown in the following sections. The maximum length of a document name is 32 characters.

The general format for received faxes and messages:
[NAME | NUMBER | ADDRESS] [SPACE + ORDER_NUMBER].

Name, number or address length is not set; if the name is too long it will be cut from the end. The order number consists of '#' and a number. Four characters are reserved for the number. The maximum number of files with the same prefix is 39999.

After the document name is a date and time stamp. [DATE] and [TIME] are shown as follows:
- the date uses the format DD.MM.YY (default)
- the time uses the format HH.MM (default).

The JPEG and GIF file naming convention differs from other naming conventions. The format is shown in 8.3.

[Name].JPG
[Name].GIF

User created documents are placed under the folder "Own texts" and the documents are named by default as follows:
[DOCUMENT TYPE] [SPACE + ORDER_NUMBER] [DATE] [TIME].
- Document types are Fax, Message, Mail and Note.
- Date and time values are as in received documents.

Memos are named as Memo:
[EVENT_START_TIME] [EVENT_DATE] [SPACE + ORDER_NUMBER] [DATE] [TIME].

Documents are indexed so that any duplicate versions have ´#n´ at the end of the name
Example 1:

Business card Anders Jargeryd #3
Business card Anders Jargeryd #2
Business card Anders Jargeryd.

5.1.4 Bubbles

When creating bubbles, there are some basic rules that should be considered:

General
Size information and confirmation note.
The selection frame is as wide as the title underline.

Text in the bubble
The title is underlined and the font is bold 20. List items are normal 18 font.
The underline of the title is 40pixels (~ 7.2 mm) from the frame above.
The title begins 24 pixels (~ 4.3mm) from the left side frame.
List items begin 38 pixels (~ 6.8mm) from the left side frame.

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

Bubble arrow

The settings bubble arrow has two possible shapes. Normally the larger one is used, but if the text is too wide in the bubble, then the smaller one is used. The bubble arrow should point vertically to the middle of the settings item line and horizontally to the black arrow or left from it if the text in the bubble is very long (Figure 6).

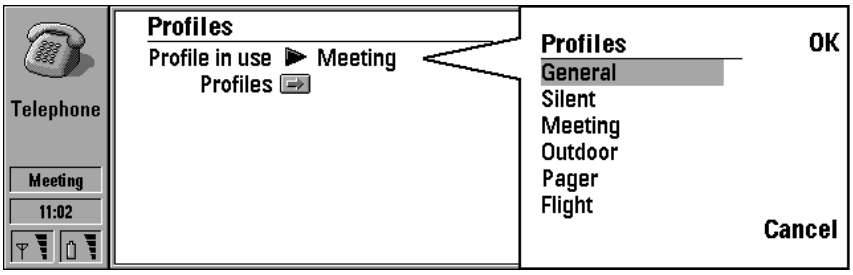


Figure 6. Bubble arrows.

5.1.5 Multiselection

Normally only one item is selected at a time and any commands that are used affect only that selected item. With multiselection functionality it is possible to select more than one item and use commands that effect all the selected items (Figure 7). Multiselection can be active in many views at the same time.

Multiselection works in all lists. It is useful for example in:

- directories,
- internet access points,
- the calendar and
- other lists.

The multiselection operation is performed in one of these ways:

- scrolling through the list by pressing Arrow keys and pressing Ctrl+Space on the selected item,
- scrolling through the list by pressing Arrow keys and pressing Shift on the selected item or
- pressing shift + scroll.

The arrow keys are used to move to the first item to be selected. Shift is pressed and held down while the remaining items in the group are selected. All items can be selected by pressing Ctrl-A. Pressing the Esc button de-selects the item that the selection bar is on.

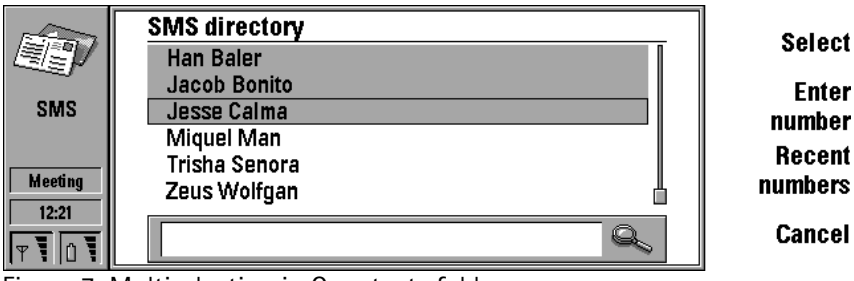


Figure 7. Multiselection in Own texts folder.

See more information about non-exclusive lists in the SDK [8] and [9].

[8] pcgeos\Techdocs\Nokia9000\Versions\Update_6.htm
[9] pcgeos\Techdocs\UI\Lists\Lists_8.htm

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

5.1.6 Settings and options

Settings and options are both set in the same way. Settings are used for permanent configurations and options for temporary configurations. There is a sample application in the SDK, which demonstrates how to build settings type dialogues for the applications [10].

5.2 Indicator, Application and Command Button areas

The Indicator area is an 80-pixel wide area containing the application icon and name. There is more information about Indicators in Chapter 14.

A frame must separate the Application area. The application area frame is a 2-pixel line with a 4-pixel wide light grey shadow (down right). The left margin from the Indicator area is 8 pixels. Inside the frame, the margins are at least 6 pixels (left and right), 5 pixels top and 4 pixels bottom.
The title font is Systemfont 20 bold, and it is located above the title line. The title line is a 1-pixel line. Under the title line is a 2-pixel space before the first list item (text). The list font is Systemfont 18 normal and seven lines fit in the list. The selection bar is light grey, 22 pixels high and as long as the title line.

In the Command Button area (CBA) the font is Systemfont 20 bold and it is recommended to use short, one line commands, but when necessary it can wrap to the next line. The right margin in the CBA is 3 pixels.

5.3. Command buttons and special keys

5.3.1 Command buttons

There are four command buttons on the right side of the display. The text labels of the buttons are located next to the buttons. The order of the command buttons should be consistent. As a basic rule the following types of commands should be assigned:

Button 1: Action button (Open, Accept, Send, Select, Go..)
Button 2: Options/ New, Add
Button 3: Delete, Clear
Button 4: Close, Cancel, Back, Hang up, Settings

All commands start with an initial case letter, only the OK command is in all-uppercase.

5.3.2 Menu button

The menu is activated from the Menu button, which is located on the left side of the display. The menu consists of commands, which are not on Command buttons (Figure 8). The contents of the menu vary according to the active view. The most important commands should be possible to activate directly from command buttons. Less important commands can be under the Menu button. If the settings item is in the menu, it is always the last item in the list.

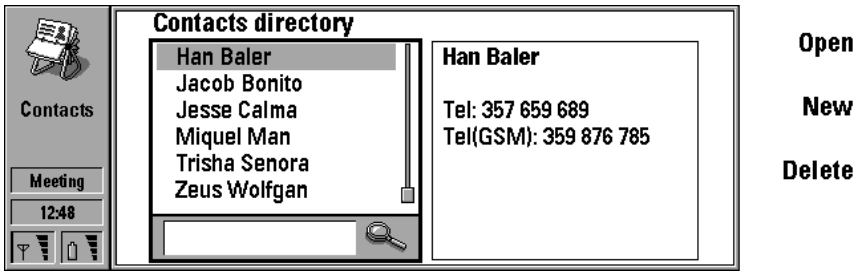


Figure 8. Menu in the Contacts view.

[10] pcgeos\N9110v10\Appl\Sdk_9000\Settings\settings.goh

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

The number of menu items is not limited. If there are more commands on the menu than fit in the view, a scroll bar appears. Pressing the Arrow keys can scroll the scroll bar and the list. Seven items can be seen on the screen at one time. If there are no menu commands for open view and the user tries to activate the menu by pressing the Menu button, a flashing note will be shown: "No menu available." (Figure 9).



Figure 9. "No menu available" note.

The Menu items are selected by pressing the Select button or the Enter key. If the user does not want to select any menu-based command, pressing the Cancel button or the Esc key can dismiss the menu. For more information, see the SDK [11]. There is also a sample application in the SDK, which shows how to display a menu [12].

5.3.3 Zoom button

Pressing the special Zoom button, which is on the left side of the display, allows the user to zoom. There are certain situations where zoom does not work:

- in Menus
- in Logs
- in settings
- in application-based notes and bubbles
- in views that use only half of the screen.

Zoom normally has three levels (see Figure 10):

- normal (100 %)
- magnified (120 %)
- huge (150 %).

In some difficult, full views the user has only two zoom levels:

- normal (100 %)
- magnified (120 %).

The zoom operation affects the middle area of the view. It doesn't affect the Screen indicator area, application title area, application icons and command button area. In the viewers and editors the zoom affects the entire appearance of the document. When the font becomes bigger, words are wrapped. The cursor stays where it was before zooming and it can always be seen (the editor takes care of this).

[11] pcgeos\Techdocs\NOKIA9000\UI\9000UI_trigger.htm
[12] pcgeos\N9110v10\Appl\Sdk_9000\Menu\menu.goh

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

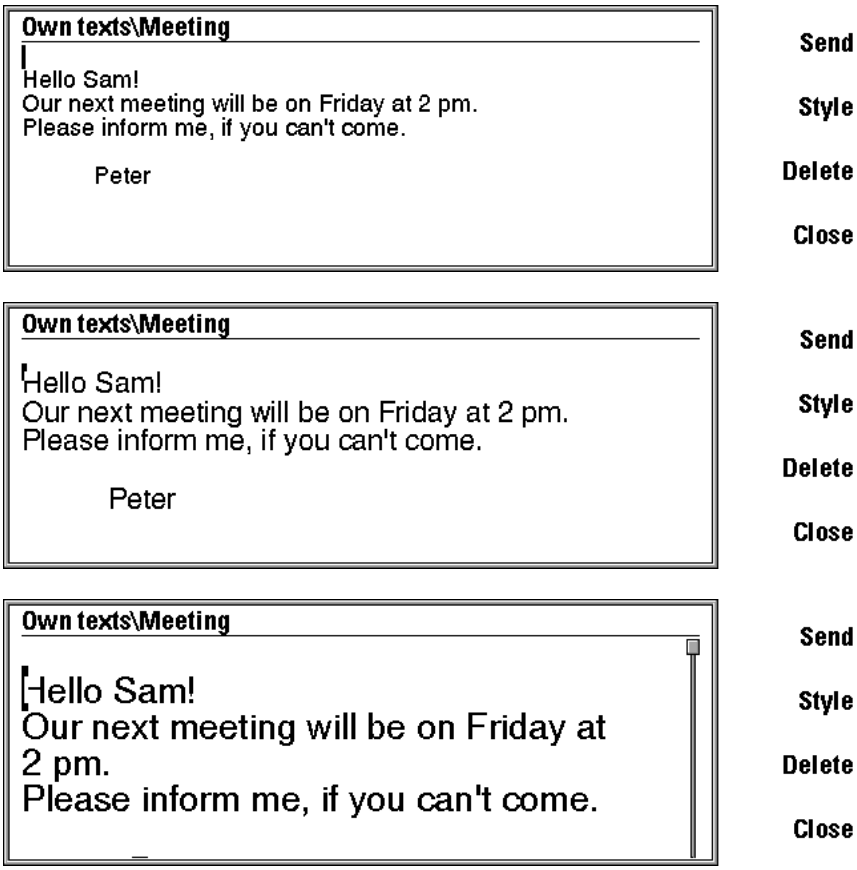


Figure 10. Normal editor, editor in 120% zoom level and 150% zoom level.

In lists, such as the Contacts directory, the size of the list item font is also increased/decreased. If the user has pressed the Zoom button in one view and changes to another view, the Zoom level is the same as in the other view. If the user had zoomed before booting, the zoom level is also saved. In those views which already have the zoom feature (e.q. opened received fax or image viewer), the functionality by pressing the Zoom button is the same as when zoomed by pressing the command buttons in those views.

When a received fax or an image is opened and the Zoom button is pressed, it can be scrolled around all seven-zoom levels. The Zoom in and Zoom out buttons are still in the Fax and Image viewers. The zooming level in Fax and Image viewers does not affect the general zoom level. It affects only to currently open file. Always when the Fax or Image viewers are started, the zoom level in the viewer is at the basic level. Please see the SDK [13] for more information.

5.3.4 Backlight button

The backlight function makes it easier to use the PDA-side in dark conditions. The user can choose the backlight period in System settings. The period can be 15/30/45 seconds or 1-5 minutes.

Necessary functions for the backlight can be found in the SDK. With these functions it is possible to turn on/off the light, see the current state of the light and set/get the backlight timeout level. APIs are defined in the respondr.goh [14] file and the foam.goh file [15].

[13] pcgeos\Techdocs\Nokia9000\UI\9000UI_zoom.htm
[14] pcgeos\N9110v10\Cinclude\Internal\respondr.goh
[15] pcgeos\N9110v10\Cinclude\foam.goh

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

5.3.5 Arrow keys

Arrow keys are used for scrolling the views horizontally or vertically. Arrow keys are used in operations, such as:

- changing from window to window (File transfer)
- when a fax is zoomed
- moving from day to day in the Calendar month view

5.3.6 Special keyboard keys

The **Esc** button is used for cancelling an operation. When a Cancel button is displayed, the alternative is to press the Esc button. Also, when an information note is on the screen, it is removed with the Esc button.
The **Enter** button is used in many applications:

- In Contacts applications (The contact card key moves the cursor to the beginning of the next row or if the Enter key is pressed on the last line of the field a new row is added to those fields).
- In the Fax application (Enter opens the selected folder or document. This feature is very common in applications where there are folders and documents).
- In the Tel application (the telephone directory or number selection bubble starts a voice call).

The **Ctrl** button is used with character keys to form shortcut keys. The Ctrl button also has a specific meaning inside the terminal emulator.
The **Shift** button allows the creation of capital letters. If pressed once and then raised, only the next character will be converted to upper case, the following character is again lower case. Together with arrow keys the Shift key is used to mark areas.
The **Caps** button allows the continuous creation of capital letters. When pressed once, all the characters will be upper case until Caps is pressed again.
The **Tab** button works like scroll down. It enables moving down from field to field. Shift + tab works like scroll up and allows moving up from field to field.
The **Chr** key functionality can be applied anywhere on the PDA-side of the communicator where the user can enter characters from the keyboard. The purpose of this operation is to allow the user to enter special letters related to different languages and some other special characters, which cannot be found directly on the keyboard (e.g. "£").

5.3.7 Shortcut keys

There are several shortcuts, which can be used in the document applications:

- Ctrl +A (selects all)
- Ctrl +X, +C, +V (Cut, Copy, Paste, respectively)
- Ctrl +Z (undo)
- Ctrl +B, +I, +U (Bold, Italic, Underlining, respectively)
- Ctrl +N, +P (opens the Next/Previous item)
- Ctrl +left/right arrow keys (moves the cursor one word backward/forward in the edit field)
- Ctrl +up/down arrow keys (moves the cursor up/down one paragraph).
- Ctrl +spacebar (marks the nearest word in all editable views and fields).
- Ctrl +backspace (deletes the previous word in all editable views and fields)
- Ctrl +H (moves the cursor and view to the beginning of the document)
- Ctrl +E (moves the cursor and view to the end of the document)
- Ctrl +T (adds the current time)
- Ctrl +D (adds the current date)
- Ctrl +M (adds the User data name)
- Ctrl +R (reloads in WWW)
- Chr +left/right arrow keys (moves the cursor to the beginning or end of the line)
- Chr +up/down arrow keys (moves the cursor a page up or down in a document)

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

- Shift +arrow keys (area marking)
- Shift +Chr +arrow up/down (selects text, page by page, inside the document)

5.4 Notes and search functionality

Different types of notes are used when there is a need to inform or warn the user about an unexpected event. It is also important to ask for confirmation if an operation will cause data to be removed and the action cannot be reversed.

This section describes only the different note types, their behaviour and appearance. Notes are classified into two groups according to their behaviour: Application and System-based notes. This document only provides information about application-based notes.

Different application notes are:

1. Information notes
2. Confirmation notes
3. Flashing notes
4. Cancel notes
5. Special notes
6. Input notes

5.4.1 Information notes

Information notes are used to inform the user that something unexpected has occurred and that the operation cannot continue. A note has an icon and the text "Note:" (see Figure 11). Other icons cannot be used. Pressing the OK button will remove the note.

Size
Height is 198 pixels (almost the whole screen), width is between 270 pixels (~ 48.6 mm) – 440 pixels (~ 79.2 mm).

Margins
There should be 22-pixel (~4 mm) top, bottom and left margins. The right margin depends on the command text (usually it is OK).

Icon
The icon is placed in the upper left corner inside the margins.

Text
The title is Bold-, italic18 ("Note:" or "Confirm:"). The text is Normal-, italic 18.

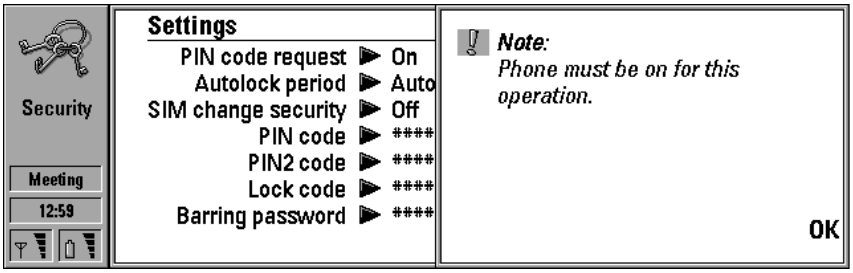


Figure 11. Information note.

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

5.4.2 Confirmation notes

Confirmation notes are used to allow users to accept or cancel operations that will remove data that cannot be undeleted. It always has an icon and the text "Confirm:" (see Figure 12). The size, margins, icons and text are the same as in the Information notes.

Cancel will remove a note and cancels the [OPERATION_NAME] operation. The [OPERATION_NAME] button will remove the note and execute the operation.
Figure 12 shows a Delete note, which also includes a wastebasket icon. Other confirmation notes do not include extra icons.

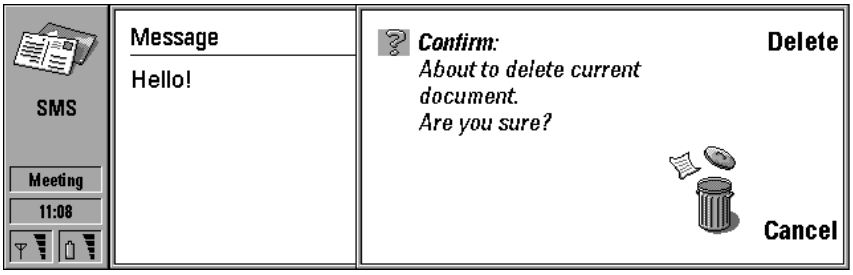


Figure 12. A delete note.

5.4.3 Flashing notes

Flashing notes show information that indicates, for example, that some operation has started and is in progress. A flashing note has no buttons, but is shown in the middle of the screen before it is removed automatically. There is an example .goh file in the SDK [16].

The time that the note is kept on the screen is either 1 or 3 seconds. The note is removed if there is any user action. There is a 0.3-second pause where no user action is accepted when the note is displayed. An operation can continue after removing the note in the background. Flashing notes are used for example when a send operation is started or cancelled. Flashing notes are also used in some error conditions, for example when a lock code is entered incorrectly (see Figure 13).

Size and placing
A flashing note is generally 183 pixels (~33 mm) high and 270 pixels (~48.6 mm) wide. If there is a problem fitting the text in the frame, it can be a little bit wider. The frame should be almost in the middle of the screen; more precisely, the frame begins 220 pixels (~ 39.6 mm) from the left edge of the screen and 7 pixels (~ 48.6 mm) from the top of the screen.

Margins
Margins should be 33 pixels (~ 6 mm); all text should be inside the margins and, if possible, any icons and animation also.

Text
Text should begin from the upper-left corner inside the margins. All text is normal-italic 18, and there is no title.

[16] pcgeos\N9110v10\Appl\Sdk_9000\Flashing\flashing.goh

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

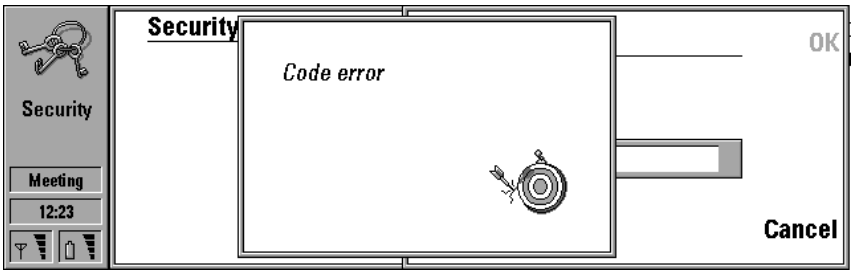


Figure 13. Flashing note.

For more information, see [17].

5.4.4 Cancel notes

Cancel notes are used during longer operations. These notes are shown on the screen until the operation is finished (see Figure 14). Typically cancel notes are used during operations such as:

- printing,
- connecting to the Internet or to remote a mailbox,
- making a request to switch centres,
- many PC connectivity operations such as: connecting to a PC, transferring data, installing software.

Size
The size is the same as that for information and confirmation notes.

Margins
There should be 33-pixel (~6 mm) margins, all text should be inside the margins and, if possible, any icons and animation also.

Text
The text should begin from the upper-left corner inside the margins. All text is normal-, italic 18, and there is no title.

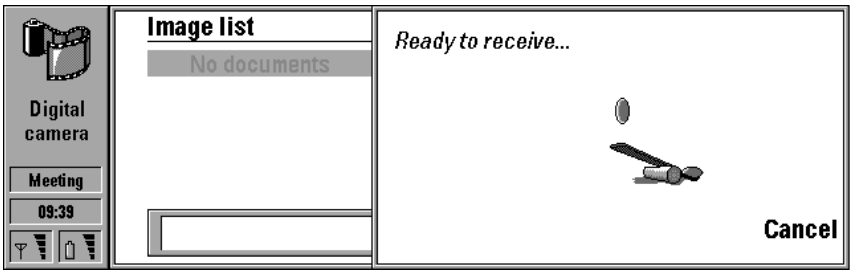


Figure 14. Cancel note.

Pressing the Cancel button terminates the current operation. Incomplete data are normally deleted. A cancel note restricts access to the current application, but the user can switch to another application. Switching to another application does not terminate the operation. A cancel note is removed if the Cancel button is pressed or when the operation is completed.

[17] Handheld Systems magazine\ May-June 1997\ Lee Ann Fujii

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

5.4.5 Special notes

Some notes are considered special notes because of their different behaviour, for example:

- a new voice call is received,
- a calendar alarm note,
- a clock alarm note,
- a new contacts note (see Figure 15).

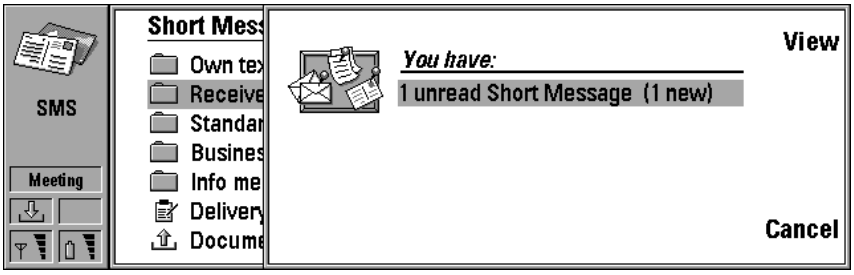


Figure 15. New contacts note.

This note is shown when the cover is opened and there are unread messages. Generally, when something new has been received a note is displayed. Unread received documents and missed calls can be opened directly from the note.

5.4.6 Input notes

Input notes are used for inserting information (see Figure 16). Examples of input notes are:

- a copy or rename note,
- an enter lock code note,
- a password code.

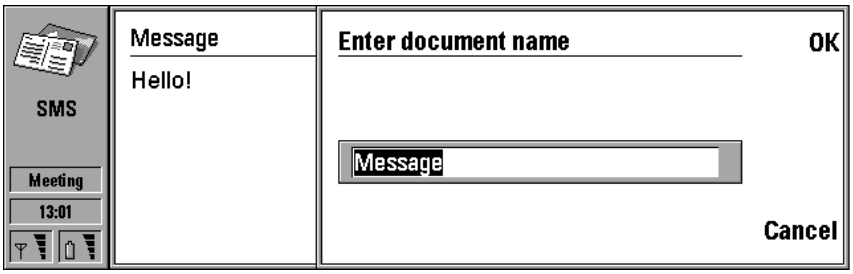


Figure 16. Input note.

Switching to another application or closing the cover will remove an input note. This cancels the operation.

The sizes of the input boxes (and also search boxes, see Chapter 5.4.7) depend on the situation. The functionality of the boxes is also different (Figure 17). The appearance of the normal input and search boxes stays the same, but text or an icon distinguishes the different functionality. The search boxes in views that have two windows differ in their appearance from the normal ones.

The width of the normal input and search box is always the same as the width of the title line. This same rule is used both in the list area and in different dialogues and notes. The magnifying glass icon is used in all search boxes that only have the search functionality. No icon is used in an input box that only has input functionality. The icon or text is always located on the right side of the box.

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

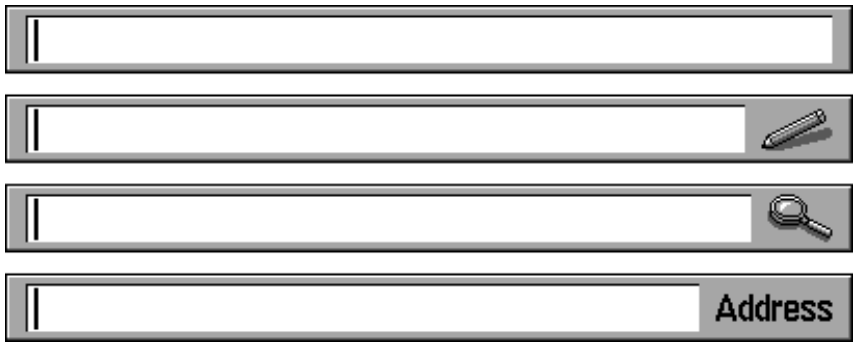


Figure 17. Different kinds of input and search boxes.

5.4.7 Search functionality

The user can perform several types of searches to find information. Entering a search string in a Search text box is one possibility. An example of a Search text box is shown in Figure 17. The search algorithm is based on an incremental search; the search results are updated dynamically in the list view as the search string changes. In the name lists, the user can search for names, companies and addresses. The needed API can be found in the SDK [18] and also some further information can be found there [19] and [20].

5.5 Graphical UI

5.5.1 Grey scales

There are 16 grey scales supported in the 9110. Grey scale mapping is linear, with 0 being black and 15 being white. However, it is recommended that bitmaps should use four greys: black, dark grey, light grey and white. Graphics can make use of the 16 Geos enumerated colour types to choose the appropriate grey. The 9110 will correctly map the colour to the correct level of grey, depending on number of screen greys available. Figure 18 displays how different colours will map to grey on 4- and 16-level grey-scale modes.

Geos colour	4 levels of grey	16 levels of grey
C_BLACK,, C_DARK_GREY	Black	Black
C_BLUE, C_LIGHT_BLUE	Black	Dark-dark grey
C_GREEN, C_LIGHT_GREEN	Dark grey	Mid-dark grey
C_CYAN, C_LIGHT_CYAN	Dark grey	Light-dark grey
C_RED, C_LIGHT_RED	Light grey	Dark-light grey
C_VIOLET, C_LIGHT_VIOLET	Light grey	Mid-dark grey
C_YELLOW, C_BROWN	White	Light-light grey
C_WHITE, C_LIGHT_GREY	White	White

Figure 18. Grey scales.

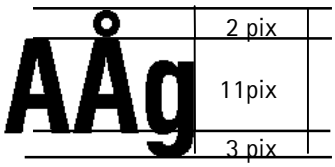
[18] pcgeos\N9110v10\Cinclude\foam.goh
[19] pcgeos\Techdocs\Nokia9000\Versions\Update_8.htm
[20] pcgeos\Techdocs\Nokia9000\UI\filter.htm

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

5.5.2 General fonts

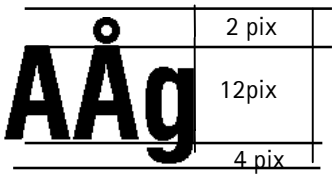
The main rules concerning font sizes:

System font size 14



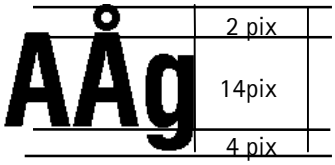
Uppermost ascender line
Ascender line
16 pix
Base line
Descent line

System font size 16



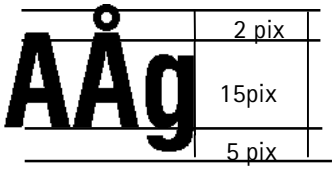
Uppermost ascender line
Ascender line
18 pix
Base line
Descent line

System font size 18



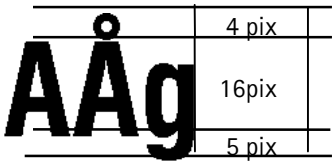
Uppermost ascender line
Ascender line
20 pix
Base line
Descent line

System font size 20



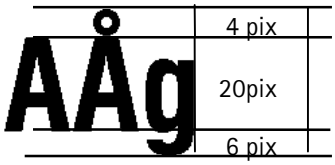
Uppermost ascender line
Ascender line
21 pix
Base line
Descent line

System font size 21



Uppermost ascender line
Ascender line
25 pix
Base line
Descent line

System font size 26



Uppermost ascender line
Ascender line
30 pix
Base line
Descent line

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

Normal Terminal font

The character size is 7 * 8 pixels, including the spacing between the characters and lines.

The total width is 80 characters * 7= 560 pixels.
The total height is 24 lines * 8= 192 pixels.

Zoomed Terminal font

The character size is 7 * 15 pixels, including the spacing between the characters and lines.

The total width is 80 characters * 7= 560 pixels.
The total height is 13 lines * 15= 195 pixels.

System fonts

Size	Style	Description
14	Bold	Text in indicator mode slots
16	Normal	Text in tight tables, e.g. general and individual logs, document outbox
16	Bold	Application names in indicator field
18	Normal	Text body
18	Normal-italic	Notes text
18	Bold	Item titles in Contact cards and in Calendar month view, name on the details view of contacts
18	Bold-italic	Note titles
18	Light	Company names, dates and times in received messages, notes, own text list etc. views
20	Bold	Main titles and command texts
21	Normal	In the 120% zoom mode (when 18-Normal font in the normal mode)
21	Bold	In the 120% zoom mode (when 18-Bold font in the normal mode), the days which have events
21	Light	In the 120% zoom mode (when 18-Light font in the normal mode)
26	Normal	In the 150% zoom mode (when 18-Normal font in the normal mode)
26	Bold	In the 150% zoom mode (when 18-Bold font in the normal mode)
26	Light	In the 150% zoom mode (when 18-Light font in the normal mode)

Table 1. System fonts.

5.5.3 General icons

Table 2 lists some common icons and their sizes. For more information, see the SDK [21]. In the SDK you can also find an Icon editor application, which allows you to create icons for applications [22].

[21] pcgeos\N9110v10\CInclude\foam.goh
[22] pcgeos\Techdocs\Tools\IconEd\index.htm

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

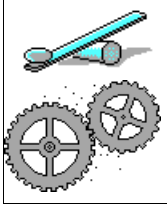
Bitmap	Description	Size
	Black arrow icon in settings view	25x18
	Grey arrow icon in settings view	25x18
	Framed arow icon in settings view	25x18
	“More” icon in command button area	25x18
	“Black” icon in command button area	25x18
	Folder list icon	25x18
	Memory card list icon	25x18
	Delivery report list icon	25x18
	Document inbox list icon	25x18
	Document outbox list icon	25x18
	Information note icon	32x20
	Conformation note icon	32x20
	Pen in search field icon	34x23
	Magnifying glass icon in search field	31x23
	Flashing note accepted icon	64x50
	Flashing note error icon	64x50
	Delete note icon	55x70
	Progress animation 1	98x71
	Cancel animation 1	98x71

Table 2. General icons list.

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

6 TELEPHONE

The SDK is not intended for programming telephone applications. It is, however, possible to launch a voice call using another application. The function can be found from the eci.goh file [23]. There is more information about the Telephone application in the Nokia 9110 Communicator user guide.

Some general principals:

- The phone application is started after device boot.
- The phone application is used to control voice calls and display information about ongoing calls.
- Calls can be controlled either from the phone panel or from the phone application.

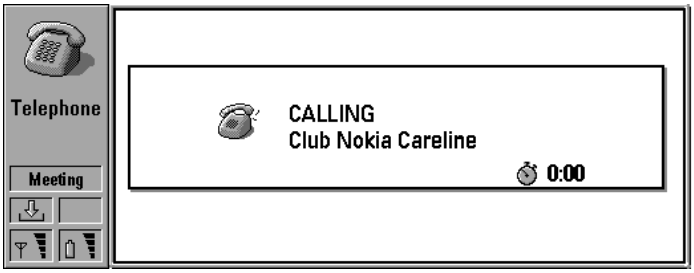


Figure 19. Example of incoming call indication.

An API exists, which enables an application to send a message from the PDA-side to the CMT-side. This feature, called **Phone display message**, is useful for example when using the 9110 in a car holder.

The function required for this can be found in the SDK, please see the eci.goh file [24].

[23] pcgeos\N9110v10\Cinclude\internal\resp\eci.goh
[24] pcgeos\N9110v10\Cinclude\internal\resp\eci.goh

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

7 FAX

The Fax application is used for viewing received faxes and sending created faxes. Fax sending and receiving has the following restrictions: sending and receiving cannot be performed if there is an active voice, data or another fax call going on. If any of these actions is active, the received fax call is rejected or the fax is put into the outbox to wait for the end of the current call.

The SDK describes how an application can use the fax libraries and Mailbox library to compose and send a fax. There is also information about how to write a program that composes a custom fax using graphics commands in the SDK [25] and [26].

From the fax main view the user can access two document folders (own texts and received faxes), create new faxes and change fax-related settings. The document outbox can be opened from the fax main view, see Figure 20.

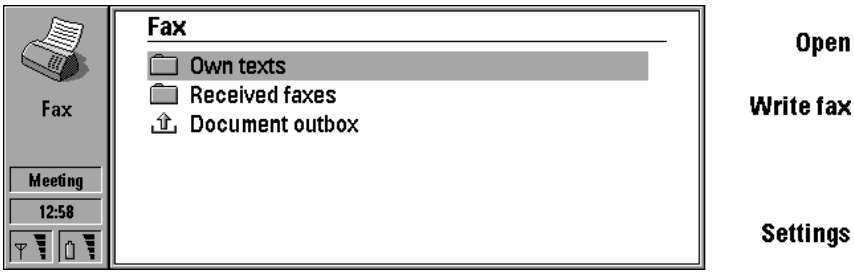


Figure 20. Fax main view.

If the connection fails during the fax sending operation, the fax remains in the outbox and is sent again when the connection has been re-established. If fax receiving fails in the middle of the operation, the fax is saved normally as a document, but information is included with the document that receiving has failed and that the document is incomplete.

7.1 Sending faxes

There is a sample application in the SDK, which demonstrates how an application can send faxes [27]. The user can send faxes using either the Fax application or the Notes application.

When the Forward or Recipient button is pressed, the fax address sheet is opened (see Figure 21). The fax address sheet supports three ways of creating recipient information:

1. Information is typed in manually.
2. Information is fetched from the Fax directory.
3. Information is fetched from the recent numbers list.

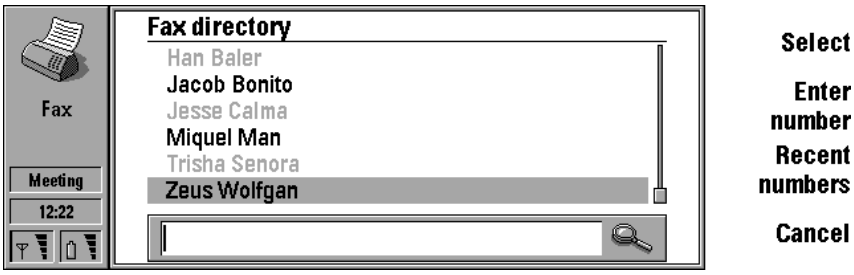


Figure 21. Fax address sheet.

[25] pcgeos\Techdocs\Communication\FaxSend\index.htm
[26] pcgeos\Techdocs\Communication\Mailbox\index.htm
[27] pcgeos\Nokia9110v10\Appl\Sdk_9000\FaxSend1\faxsend1.goh

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

7.2 Some fax settings

The TTL header is one line at the top of each page. It includes the following information:

1. Date and time of transmission.
2. Name and fax number of the sender.
3. Page number.

The sender name and number are taken from the user data. If the user data does not contain the necessary information, “Nokia 9110 Communicator” will be used.

The Settings view includes all the fax application related permanent settings. Settings are valid after they are changed in the settings list.

7.2.1 Send time

This setting allows the user to choose when the fax will be sent. This is a useful feature because sometimes the user wants to send the fax at a certain time. The default value is Immediately. The user can set a new default time by typing a time (Time format is hh:mm).

7.2.2 Logo file

This setting enables or disables the use of a Logo. The logo file can be set as No image or a file name can be selected from the list. The list shows all GIF and JPG files from the downloaded files folder. By selecting a file, the user can place a logo on the cover page.

7.2.3 Signature file

This setting enables or disables the use of a signature. The signature file has to be stored as a GIF or a JPG file in the downloaded files folder. The signature file selection can be set to No image or a file name can be selected from the list. By selecting one file the user can place a signature on the cover page. The signature is located under the remarks field.

7.2.4 Resend

If sending of a fax fails, the Outbox will try to send the fax again. With this option, it is possible to define which pages are sent again. The cover page shows which pages are sent again, but the TTL header treats each fax as a new one.

7.2.5 ECM

This setting enables or disables Error Correction Mode. The default value is: "On".

7.2.6 Fax polling

When the state is set to "On", the polling fax call begins. This call is direct (not handled by the Outbox).

7.3 Receiving faxes

Faxes are received in the background and the fax call indicator is switched on. When the new fax is received, the user is informed with a flashing note. The receive fax operation can be cancelled by pressing “end call” on the phone interface or by initiating a voice call from the Phone application.

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

8 SMS

This section describes the common features of the SMS application user interface. The Short Message Service lets the user send and receive messages to and from the GSM network. The application attempts to deliver messages to a Mobile Station (MS), whenever it is registered on the network (even when the MS is engaged in a voice or data call). An MS may also roam the network and messages can be sent and received. For more information see [28].

8.1 SMS Main view

The SMS main view is used to open folders, to write new messages and to change message-related settings (Figure 22). Also the Delivery report option is shown in the main view. If the radio unit does not have a connection to a service centre, a message can stay in the outbox until the connection is established.

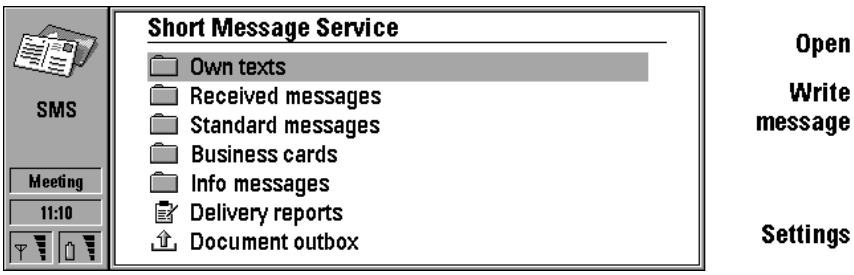


Figure 22. SMS main view.

8.2 Received messages

New messages are received automatically and placed in the Received messages folder. A received message can be one of the following types:

- a normal message (as in Figure 23),
- a normal message stored in the SIM card,
- a business card,
- a service card or
- notifications of new voice messages from the voice mailbox.

The message is presented in the form of a postcard, where a business card or service card may be attached with a paper clip. Received messages have a stamp with a postmark on the card.

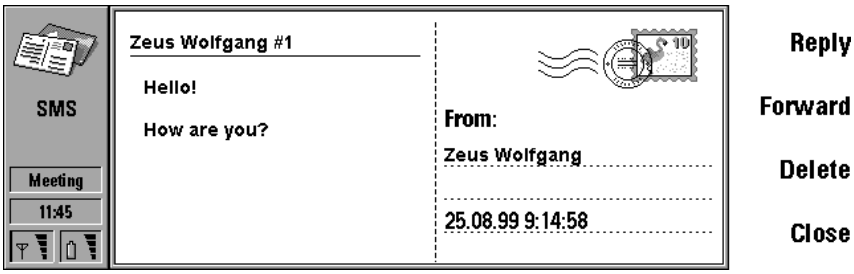


Figure 23. Opened received short message.

The GSM standard requires that SIM-specific messages must be transferred to the SMS data field in the SIM card. These messages can be read and deleted using the SMS application. The functionality is the same as that of a normal message.

[28] Geoworks Nokia 9000 Communicator Programming Class Notes\SMS Messaging

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

8.3 Create and send a message

The message is written into an empty postcard. When the Forward or Recipient button is pressed, the fax address sheet is opened (see Figure 24).

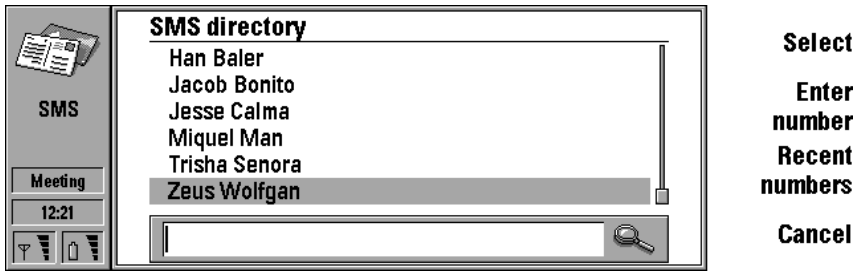


Figure 24. SMS address sheet.

By sending an electronic business card the user can send his or her user data or any contact entry to GSM-based equipment that supports SMS. There is an item for the user's card and all the entries from the Contacts directory in alphabetical order. Business cards cannot be edited or deleted from the list.

For more information about SMS sending operations see the SDK [29] and [30]. Documentation for making an SMS header is also located in the SDK [31].

8.4 Concatenated short messages

The GSM specification version 03.40 defines concatenated short messages, which can be longer than 160 characters. If the message is long (over 160 characters), it will be sent in pieces. The user is informed with the note. Nokia 9110 Communicator supports messages up to 2280 characters.

It is necessary to use a specific header with concatenated short messages. This header defines the message so that several messages can be patched into one. The header is 56 bits long so it includes 7 bytes (see Figure 25):

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte
Length of the header	IEI	Length of the sub-header	Reference number	Maximum amount of pieces	Sequence number	Fill

IEI=Information Element Identifier

Figure 25. Concatenated SMS header.

The other way for sending long messages is to use NBS headers. The idea is to include a text-based Narrow Band Socket (NBS) header with the message. Detailed information about this can be found at Forum Nokia [32].

[29] pcgeos\Techdocs\Nokia9000\SMS\index.htm
[30] pcgeos\N9110v10\Appl\Sdk_9000\Smssamp\smssamp.goh
[31] pcgeos\Techdocs\Nokia9000\Versions\Update_4.htm
[32] http://www.forum.nokia.com (ssm2_0_0.pdf)

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

9 INTERNET

This chapter describes the common features of the user interface in the Internet application. After pressing the Internet application button, the Internet service list is shown (as in Figure 26).

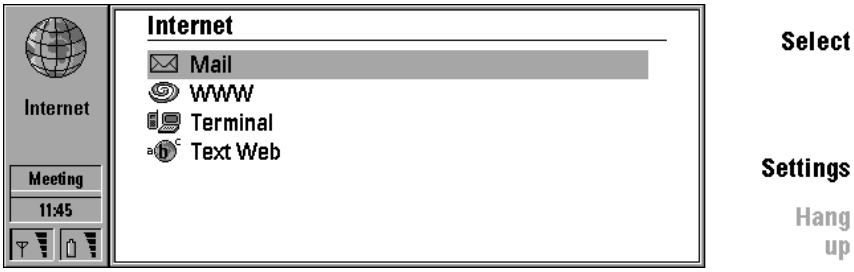


Figure 26. Internet display

9.1 Mail

After selecting the Mail application from the Internet display, the Mail view is opened (Figure 27).

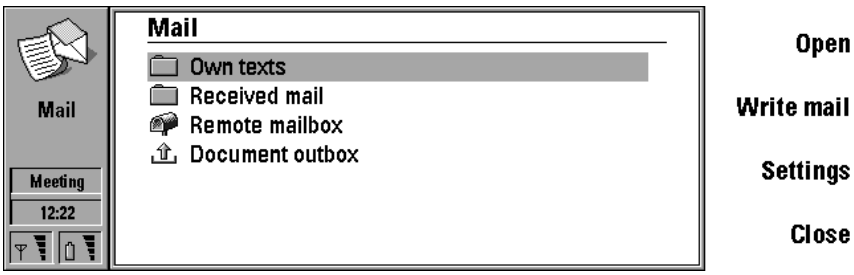


Figure 27. Mail main view.

There is a sample application in the SDK, which demonstrates how to construct, register and send an e-mail message [33]. The SDK does not offer an interface for receiving mail, but it is possible to create receiving applications using the socket API. Documentation and API can be found in the SDK [34].

After selecting a Remote mailbox, it is possible to connect to it. The size of the mail message is given in kilobytes. The protocol between the Nokia 9110 Communicator and the Remote mailbox is IMAP4 or POP3.

9.1.2 Building header information

When sending mail without a mail address controller, two custom structures need to be defined for the message header.

```
/* Message envelope */
typedef struct {
  VMChainTree   vmt;          /* structure starting a tree block */
  word          link_id;      /* mail transport driver internal */
  dword         return_path;  /* error return address list   */
  dword         from;         /* originator address list */
  dword         sender;       /* sender address list */
  dword         reply_to;     /* reply address list */
  dword         to;           /* primary recipient list */
  dword         cc;           /* secondary recipient list */
```

[33] pcgeos\N9110v10\Appl\Sdk_9000\Email\email.goh
[34] pcgeos\Techdocs\Communication\Socket\Socket_a.htm

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

```
  dword         bcc;                /* blind secondary recipient list */
  DBGroupAndItem remail;            /* remail header if any list   */
  DBGroupAndItem date;              /* message composition date string */
  DBGroupAndItem subject;           /* message subject string */
  DBGroupAndItem in_reply_to;       /* replied message ID */
  DBGroupAndItem message_id;        /* message ID */
  DBGroupAndItem newsgroups;        /* USENET newsgroups */
  DBGroupAndItem sending_host       /* mail transport driver internal*/
  VMChain next;                     /* pointer to next address block */
} ENVELOPE;

/* Item in an address list */
typedef struct {
  DBGroupAndItem personal;          /* personal name phrase */
  DBGroupAndItem adl;               /* at-domain-list source route */
  DBGroupAndItem mailbox;           /* mailbox name */
  DBGroupAndItem host;              /* domain name of mailbox's host */
  DBGroupAndItem error;             /* error in address */
} ADDRESS;
```

The message header is stored in the VM block, which is allocated from the mailbox VM file. Header items are allocated as ungrouped DB items from the same file; items, that are not used, are set to 0.

A block must start with ENVELOPE structure. ADDRESS structures are placed in the same block immediately after the ENVELOPE. The index of the first address in the address list is stored in the ENVELOPE structure fields, so all addresses of the same type must be stored subsequently. If the address list contains no addresses, index 1 is used. A from address and at least one recipient are required. A mailbox and host are required in each address. Unused fields are set to 0.

This example allocates and initialises a header block containing two ADDRESS-structures. If space for more addresses is allocated, the number of links in the VMCT_count needs to be incremented.

```
VMFileHandle        vmFile;         /* Handle of VM file for header */
VMBlockHandle       vmBlock;        /* Handle of VM block for header */
ENVELOPE            *env;           /* Pointer to ENVELOPE structure */
ADDRESS             *adr;           /* Pointer to ADDRESS structure */
MemHandle           memRef;         /* Handle for locked VM block */
word                status;         /* VMOpen status code */

vmFile = MailboxGetVMFile(0, &status);
vmBlock = VMAlloc(vmFile, sizeof(ENVELOPE) + 2 * sizeof(ADDRESS), 0);
env = (EN7VELOPE *) VMLock(vmFile, vmBlock, &memRef);
env->vmt.VMCT_meta.VMC_next = VM_CHAIN_TREE;
env->vmt.VMCT_offset = offsetof(ENVELOPE, remail);
env->vmt.VMCT_count = 8 + 2 * 5;
env->return_path = env->from = env->sender =
env->reply_to = env->to = env->cc = env->bcc = 0xffffffff;
env->link_id = 0;
env->remail = 0;
env->date = 0;
env->subject = 0;
env->in_reply_to = env->message_id = env->newsgroups = 0;
env->sending_host = env->next = 0;
```


Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

```
addr = (ADDRESS *) (env + 1);
addr->personal = addr->adl = addr->mailbox = addr->host = addr->error = 0;
addr++;
addr->personal = addr->adl = addr->mailbox = addr->host = addr->error = 0;
VMDirty(memRef);
VMUnlock(memRef);
```

Passing the header block to the mail transport driver is done using a custom structure:

```
typedef struct {
  FileLongName      EITD_fileName;          /* Name of header file */
  VMFileHandle    EITD_VMFileHandle;        /* Handle of header file */
  VMBlockHandle EITD_VMBlockHandle;         /* Handle of header block */
  word              EITD_MimeEncoding;      /* MIME character encoding */
  word              EITD_numFiles;          /* Number of attached files */
} EnvelopeInTransData;
```

The structure contains two fields in addition to header identifying information. The first field defines encoding; if non-zero, the message text will be encoded as quoted printable. The second defines the number of attached files, which has to be zero, if there are no attachments included.
The structure is allocated as an ungrouped DB item from the mailbox admin file, which can be retrieved by the MailboxGetAdminFile() function. This item is passed to the mailbox library when registering the message. The mail transport driver takes care of freeing allocated space after transmission.

```
VMFileHandle                adminFile;
DBGroupAndItem              transData;
EnvelopeInTransData *data;

  adminFile = MailboxGetAdminFile();
  transData = DBAllocUngrouped(adminFile, sizeof(EnvelopeInTransData));
  data = (EnvelopeInTransData *) DBLockUngrouped(adminFile, transData);
  MailboxGetVMFileName(vmFile, &data->EITD_fileName);
  data->EITD_VMFileHandle = vmFile;
  data->EITD_VMBlockHandle = vmBlock;
  data->EITD_MimeEncoding = 0;
  data->EITD_numFiles = 0;
  DBDirty(data);
  DBUnlock(data);
```

9.1.3 Registering a message in the mailbox

Registering is done by the MailboxRegisterMessage() function after filling the mailboxRegisterMessageArgs structure. In this structure a pointer to the message body and other information are passed to the mailbox library. For more information, see the SDK[35].

MRA_bodyStorage
  Message body storage. For the mail transport driver this is set to {
  GMSID_VM_TREE, MANUFACTURER_ID_GEOWORKS }.

MRA_bodyFormat
  Message body data format. This is set to {
  GMDFID_TEXT_ARRAY, MANUFACTURER_ID_GEOWORKS }

[35] pcgeos\N9110v10\CInclude\mailbox.goh

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

Indicating that we are sending a text message.

MRA_bodyRef
Pointer to the address of the body data. The format of the address is VMTreeAppRef, when GMSID_VM_TREE is used. Actual data are stored to HugeArray:

```
VMTreeAppRef myAppRef;
vmFile = MailboxGetVMFile(0, &status);
vmBlock = HugeArrayCreate(vmFile, sizeof(TCHAR), 0);
myAppRef.VMTAR_vmFile = vmFile;
MRA_bodyRef = &myAppRef;
```

MRA_bodyRefLen
The number of bytes in the body's address, i.e. sizeof(VMTreeAppRef).
MRA_transport
The means by which the message will be sent { GMTID_SMTP, MANUFACTURER_ID_GEOWORKS }.

MRA_transOption
The transport option within the selected transport driver. This is set to 0 for the mail transport driver.

MRA_transAddrs
This field is used to pass the name or IP address of the sending host to the mail transport driver.

MTA_transAddr
The first word in the buffer pointed by MTA_transAddr is the ID of the Internet access point, which can be retrieved from the access point controller. This identifier is immediately followed by a host name. The name must be null terminated.

MTA_transAddrLen

The number of bytes in the address, i.e. sizeof(word) + length of host name + 1.

MTA_userTransAddr
The user-readable form of the recipient will be shown in the Document outbox.

MRA_numTransAddrs
Number of entries in MRA_transAddrs. This is set to 1.

MRA_transData
Data for the transport driver. This field is used to pass the header information described above.

MRA_flags
Control flags. These are described in: pcgeos\N9110v10\CInclude\mailbox.goh.

MRA_summary
Description of the message, which will be shown in the Document outbox.

MRA_destApp
Application which will receive the message. This can be freely chosen.

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

MRA_startBound
The time after which a message can be sent. Pass MAILBOX_NOW to make the message available for sending immediately.

MRA_endBound
Time before which the message must be sent. Pass MAILBOX_ETERNITY to have no deadline.

9.1.4 Setting up TCP/IP Internet services

More information about this can be found in the SDK [36] and Forum Nokia [37].

The user can find these features from the Internet Access Point settings:

- Login customisation (none/manual/script) and
- PPP compression which supports Stac LZS (the compression can be in on or off mode).
- MPPS (Microsoft Point-to-Point Compression)

Connection time-outs

There are no application-specific connection timeouts. There is one time-out specified in Internet settings that governs all the TCP/IP applications. If all the open data connections have not been used for at least the time specified by this setting, the data connection will be closed and during this operation a flashing note is shown with the text: "Disconnecting from the network due to time-out..."

9.2 WWW

The browser supports HTML 3.2 with forms. The width of the hypertext viewer is 540 pixels. The total width of the view is 640 pixels. The left side arrows require 25 pixels and the right side buttons 75 pixels.

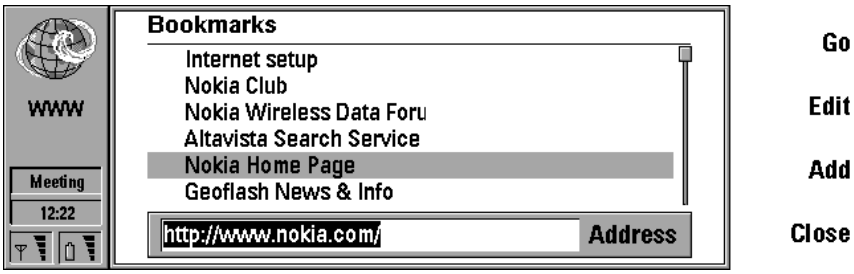


Figure 28. WWW main view.

The WWW main view contains a file selector showing the opened Bookmark list in the left pane, the URL entry box in the bottom and command buttons in the right pane (Figure 28). The first item in the Bookmark list is a link to the last visited document from the previous session. The size of the Bookmark list is unlimited and multiselection is possible. The form data length in the submit string is 2000 characters. Bookmarks can be added and edited in the main view. For more information about the WWW, please see Forum Nokia [38].

9.2.1 Inline and External image viewers

Unfetched images and image maps are displayed as image icons in the text as any character (see Figure 29). The image viewer is part of the hypertext viewer and is used when an inline image or a JPEG or GIF is fetched into the document by user initiation or with autoload images turned on.



Figure 29. Image map and inline image icons.

[36] pcgeos\N9110v10\Appl\Sdk_9000\Talk\talk.goh
[37] http://www.forum.nokia.com (sdialin.pdf)
[38] http://www.forum.nokia.com (sbrowser.pdf)

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

An external image is displayed in the image viewer view separately from the hypertext document. This view is entered automatically from the main view/Menu button/History view when a JPEG or GIF image is fetched by some link.

Image view, zoom functionality and outlook are the same as in Fax viewing (see Chapter 7). By pressing the Copy image button in the image viewer, the viewed item will be moved to the downloaded files folder using its server given name. During the copy operation, a flashing note "Copying..." is displayed.

9.2.2 How to start image viewer inside an application

This sample program will open the image viewer the same way as the WWW application does in External image.

```
@chunk TCHAR       APPL_FILE_TOKEN[] =                    "APL";
@chunk TCHAR       IMAGEV_FILE_TOKEN[] =      ".JPG";

#define MANUFACTURER                          11
#define FILE_MAJOR_PROTOCOL     1
#define FILE_MINOR_PROTOCOL     0
#define TOKEN_CHARS_LENGTH      4

void loadImageViewer (DocumentInfo name )
{
/* Open image viewer with given imagefile name */
  MemHandle localHeap = NullHandle;
  ChunkHandle paramsHandle = NullHandle, libNameHandle = NullHandle;
  TokenChars token;
  FileLongName *libName= NULL;
  ViewerParams *vpViewer= NULL;
  FileAttrInfo fileAttr;   /* File attributes of new doc */
  hptr  imageLibHandle= NullHandle;
  optr imageLibUI= NULL;
  TCHAR * string = NULL;

/* Create block for local heap. */
  localHeap = MemAlloc(TWO_KILO_BYTES,
  HF_SWAPABLE | HF_SHARABLE,
  HAF_ZERO_INIT | HAF_LOCK);

  if (localHeap != NullHandle)
  {
  /* Initialize local heap. */
  LMemInitHeap(localHeap, LMEM_TYPE_GENERAL, 0, sizeof(LMemBlockHeader),
  STD_INIT_HANDLES, STD_INIT_HEAP);

 /* Allocate chunks. */
 paramsHandle = LmemAlloc(localHeap, sizeof(ViewerParams));
 libNameHandle = LmemAlloc(localHeap, sizeof(FileLongName));

 /* Create pointers to chunks. */
 vpViewer = (ViewerParams *) LMemDerefHandles(localHeap, paramsHandle);
 libName = (FileLongName *) LMemDerefHandles(localHeap, libNameHandle);

 /* Set general document info. */
```


Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

```
    MemLock(OptrToHandle(@APPL_FILE_TOKEN));
    string = LMemDeref(@APPL_FILE_TOKEN);
    strcpy(fileAttr.FAI_token.GT_chars, string);
    MemUnlock(OptrToHandle(@APPL_FILE_TOKEN));

    fileAttr.FAI_token.GT_manufID =  MANUFACTURER;
    fileAttr.FAI_protocol.PN_major = FILE_MAJOR_PROTOCOL;
    fileAttr.FAI_protocol.PN_minor = FILE_MINOR_PROTOCOL;

    /* Copy IMAGEV_FILE_TOKEN from chunk */
    MemLock(OptrToHandle(@IMAGEV_FILE_TOKEN));
    string = LMemDeref(@IMAGEV_FILE_TOKEN);
    StrXCpy(token, string, TOKEN_CHARS_LENGTH+1);
    MemUnlock(OptrToHandle(@IMAGEV_FILE_TOKEN));

    ViewerGetLibraryNameFromToken(&token, libName);
    strcpy(vpViewer->VP_libraryName, libName);
    vpViewer->VP_documentInfo = name;
    vpViewer->VP_app = fileAttr.FAI_token;
    vpViewer->VP_destination = @SomeObject;
    vpViewer->VP_closeMsg = MSG_CLOSE_IMAGE_VIEWER;
    vpViewer->VP_nextPrevMsg = 0;
    vpViewer->VP_attr = VA_NEW_DOCUMENT;
    vpViewer->VP_state = 0;

    imageLibHandle = ViewerLibraryOpen(*vpViewer, &imageLibUI);

    @call SomeObject::MSG_SAVE_IMAGE_HANDLE (imageLibHandle);
    MemFree(localHeap);
    }
  }
```

9.2.3 Add-on software downloading view

This view is entered when an add-on software type item is fetched; a hyperlink pointing to the add-on software server is traversed.

The Install command installs the software in the Nokia 9110 Communicator. During the installation process there is a flashing note "Installing <item name>...". If CRITICAL_LEVEL is reached during installation, there is a confirmation note "Insufficient memory to download file..." By pressing Cancel the install operation is cancelled and the partly loaded software is removed from local storage.

9.2.4 How to develop a WWW plug-in

Plug- in geode

A plug-in can be an application or a library. If a plug-in is a single launchable geode, it must use MSG_META_QUIT to terminate itself to be able to launch multiple times.

The plug-in application geode must be installed in the WORLD directory (the plug-in cannot be launched by the user) or WORLD\EXTRAPPS (the plug-in can be launched by the user from the Extras menu) directory and the plug-in library geode in the SYSTEM directory.

How a plug-in is bound to the WWW application

The WWW application needs to know several things about the plug-in to be able to use it correctly:

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

- Plug-in name
- Plug-in geode name
- MIME type that plug-in can handle
- File extensions of plug-in data files.

This information is saved in an .ini file during plug-in installation. When the WWW application encounters unsupported MIME (or data file), it checks if any of the plug-ins present can handle it. If a suitable plug-in is found, it will be launched and data are passed to it.

Only one MIME type and file extension should be bound to one plug-in at a time. However, if you need to bind more than one file type to one plug-in, write a separate plug-in info file for each type containing appropriate .ini file entries.

Installing a plug-in
Plug-ins can be installed currently with WWW Application Installer. This installer needs:
- the actual geode (tiffshow.geo)
- the plug-in info file (tiffshow.inf)
- the installer file (tiffshow.ins)

These files are packed with TAR to one file, which has the extension .aos (tiffshow.aos). If you need more information concerning TAR, please contact the developer support group [39]. Also see Chapter 21 for .aos file creating instructions.

The plug-in info file
This file contains all the information a WWW application needs to use a plug-in. The file must have an .inf extension. Do not use the name plugin.inf, because others may have similar ideas; use the same name as the plug-in geode if possible. The format of the text in this file is:
<plug-in name>,<plug-in geode name>,<MIME type>,<file extension of data file>
Example:
Tiff viewer,tiffshow.geo,IMAGE/TIFF,.tif

Remember the dot in the extension and do not use line breaks.
This file must be installed in the DOCUMENT/INTERNET directory.

Installer file
This file contains information as to which directory geode and plug-in info file must be installed in the Nokia 9110 Communicator.

The format of this text file is:
GEOS Package Description File v1.0
<linebreak>
<description 1>
<description 2>
<size of geode file>
.
<plug-in geode name>
<plug-in geode name with path, where it will be installed>
<plug-in inf-file name>
<plug-in inf-file name with path, where it will be installed>
An example of the WWW Application Installer's .ins-file:
GEOS Package Description File v1.0

[39] 9110.sw.developer@nokia.com

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

```
Tiffshow
TiffViewer
3634
.
tiffshow.geo
world\extrapps\tiffshow.geo
tiffshow.inf
document\internet\tiffshow.inf
```

How a plug-in gets the data file

Add the following code to the MSG_GEN_PROCESS_OPEN_APPLICATION method of your plug-in application.

```
<plug-in process class>, MSG_GEN_PROCESS_OPEN_APPLICATION
{
        AppLaunchBlock          *myAppLaunchBlock;
        FileHandle              fhDataFile;

        /* lock the launchblock and get a pointer to it */
        myAppLaunchBlock = MemLock(launchBlock);

        /* save current directory to stack */
        FilePushDir();

        /* change to data files directory */
        FileSetCurrentPath(SP_TOP, myAppLaunchBlock->ALB_path);

        /* open data file */
        fhDataFile = FileOpen(myAppLaunchBlock->ALB_dataFile,

        FILE_ACCESS_R|FILE_DENY_NONE);
        if(fhDataFile)
        {
                        /* data file opened successfully */
}
else
{
                        /* data file could not be opened */
}

        /* unlock the launchblock */
        MemUnlock(launchBlock);

        /* restore directory from stack */

        FilePopDir();

        /* remember to call default action */
        @callsuper();
}
```


Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

9.2.5 WWW application´s external interfaces

The interface between the WWW and other applications is implemented by using GEOS Inter-Application Communications Protocol (IACP). When some application requires a WWW application to do something, a message with suitable parameters is sent via the IACP.

There are three phases when a message is sent via the IACP to the WWW:
1. Open an IACP connection to the WWW Application
2. Send the message
3. Close the IACP connection.

The WWW application has three operations, which can be performed by sending messages via the IACP:
1. Waking up WWW / Bringing the WWW to the top
2. Adding to the Bookmark list
3. Automatic document fetching.

If you want more information about the phases and operations, please use the Developer Discussion Forum [40] or contact the developer support group [41]. For more information about the WWW SMS Forms extension, see Forum Nokia [42].

9.2.6 WWW Application installer

This section specifies how to use the Nokia 9110 Communicator WWW browser to download, install and automatically run a program.

When the Nokia 9110 WWW Browser is used to fetch a file with a MIME type application/x-nokia-9110-Communicator-add-on-software, the WWW Application Installer is launched to handle it. The file in question is an AOS package and contains files to be installed in the Nokia 9110 Communicator. The files can be new applications, drivers or libraries, updates or data files. Usually files are just installed, but this document describes how one can automatically run an installed application after installation. This functionality can be used for example with a separate installer or registration program.

Package Description File (see also Chapter 21)
The program itself does not need any special preparations; it is a normal GEOS application. The only thing required is to install the program in question in the PRIVDATA/STATE directory. The program is installed in the Nokia 9110, run after a completed installation and deleted at the next reboot.

A sample INS file:
GEOS Package Description File v1.0

```
Program update
29 May 1997
10000
.
update.dat
privdata\state\update.dat
update.geo
privdata\state\update.geo
```

[40] http://dev.forum.nokia.com/wwwboard
[41] 9110.sw.developer@nokia.com
[42] http://www.forum.nokia.com (ssmsforms.pdf)

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

9.2.7 WWW Browser Style guide

An important WWW Browser Style guide for the Nokia 9110 Communicator is available on the Internet. Web services, which are created according to the guidelines set by Nokia, can receive a Nokia Communicator Optimized Web Service Logo. "The Nokia Communicator Friendly Site Program" is targeted at those interested in offering optimized Web services. More information about the program and the Style guide can be found at: http://www.forum.nokia.com

9.3 Telnet, Terminal and Text Web

The **Telnet connection** display is entered from Internet services. The SDK does not offer an interface for Telnet, but it is possible to create Telnet applications using the socket API. There is more information about the Telnet application in the SDK, for example "how to launch the Telnet application" [43].

The **Terminal application** is used to directly connect (not a TCP/IP connection) to a remote computer by a data call. In general, the Terminal connection display is very similar to the Telnet connection display. The main differences are different display titles and a different connection definition display. The SDK is not intended to program Terminal applications. There is more information about the Terminal application in the user guide. The Terminal application can be configured via SMS.

The **Text Web** allows the user to access the Internet via Smart Messages. It also provides a quick access to Nokia and Operator services. There is more information about Text Web in the Nokia 9110 Communicator user guide.

9.4 General guides for developing Internet applications

There is no necessary need to understand or develop large non-standard SW modules. Only basic Socket API functions: socket create, connect, write, read and close are needed to develop an application which is able to create an Internet/Intranet/Extranet connection.

Each SW part works in its own module, and an application can be developed using a number of modules such as the Document outbox in mail sending or the Resolver for the host name resolution, for example the imaginary URL and IP address: www.forum.nokia.com > 123.123.123.12.

Program code must be ready for multitasking, so there is a lot of testing to do. Nokia 9110 Communicator shows the error messages and handles connection procedures. Third-party SW developers must take care of socket error messages. SW installation updates can be done via cable or the Internet (WWW).

Code examples are found in Chapters 9.6.1-9.6.4 for:

- resolving addresses,
- connecting to a server,
- sending and receiving,
- closing connections.

9.4.1 Resolving addresses:

```
/* Unresolved address structure passed to SocketResolve. */
typedef struct {
  TcpAccPntExtendedAddress   UTA_link;
  byte UTA_ip[MAX_IP_ADDR_STRING_LENGTH];
} UnresolvedTCPAddress;
```

[43] pcgeos\N9110v10\Appl\Sdk_9000\TelLaun\tellaun.goh

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

```
/* Full resolved address structure passed to SocketConnect */
typedef struct {
  SocketAddress RSA_sock;
  TcpAccPntResolvedAddress   RSA_ip;
  } ResolvedSocketAddress;

/* Use the TCP/IP protocol stack. */
char tcpDomain[] = "TCPIP";

/* Buffer where we will store the address supplied by the user before resolving it. */
UnresolvedTCPAddress rawAddress;

/* Buffer where resolved address will be stored. */
ResolvedSocketAddress theAddress;

/* Fill SocketAddress part of ResolvedSocketAddress. */
theAddress.RSA_sock.SA_port.SP_port = TCP_SERVER_PORT;
theAddress.RSA_sock.SA_port.SP_manuf = MANUFACTURER_ID_SOCKET_16BIT_PORT;
theAddress.RSA_sock.SA_domainSize = strlen(tcpDomain);
theAddress.RSA_sock.SA_domain = tcpDomain;

/* Store link info into raw address buffer. */
rawAddress.UTA_link.TAPEA_linkSize = 3;                /* Size of link info        */
rawAddress.UTA_link.TAPEA_linkType = LT_ID;            /* Type of link info        */
rawAddress.UTA_link.TAPEA_accPntID = id;               /* Access point id          */

/* Copy hostname into raw address buffer. */
strcpy(rawAddress.UTA_ip, hostname);

/* Resolve the raw address into ResolvedSocketAddress. */
theAddress.RSA_sock.SA_addressSize =
SocketResolve(theAddress.RSA_sock.SA_domain,
(byte *) &rawAddress,
sizeof(TcpAccPntExtendedAddress) + strlen(hostname),
(byte *) &theAddress.RSA_ip,
sizeof(TcpAccPntResolvedAddress));
if (theAddress.RSA_sock.SA_addressSize == 0) {
  sockError = ThreadGetError();
                    ...
  }
```

9.4.2 Connecting to the server:

```
/* Create a socket which we will use to identify this connection to the socket library. */
sock = SocketCreate(SDT_STREAM);
if (sock == 0) {
  sockError = ThreadGetError();
                    ...
}

/* Dial the phone and get a PPP connection. */
sockError = SocketOpenDomainMedium((SocketAddress *) &theAddress,
CONNECT_TIMEOUT);
```


Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

```
/* Make a TCP connection. */
sockError = SocketConnect(sock,
(SocketAddress *) &theAddress,
CONNECT_TIMEOUT);
```

9.4.3 Sending and receiving:

```
/* Send and receive data. Address is not needed for stream sockets. */
sockError = SocketSend(sock,
sendBuf,
size,
0,
(SocketAddress *) 0);
size = SocketRecv(sock, recvBuf, sizeof(recvBuf), RECEIVE_TIMEOUT, 0,
(SocketAddress *) 0);
if (size == 0) {
  sockError = ThreadGetError();
                          ...}
```

9.4.4 Closing a connection:

```
/* Notify the server that no more data will be sent over this socket. */
SocketCloseSend(sock);
/* Close and destroy socket. */
SocketClose(sock);
/* Close data call using cell phone medium. */
MediumAndUnit theMedium = {
{ GMID_CELL_MODEM, MANUFACTURER_ID_GEOWORKS },
MUT_NONE,
0 };
SocketCloseDomainMedium(tcpDomain, &theMedium, FALSE);
```

9.4.5 Internet communication protocols stack

The Nokia 9110 Communicator Internet communication protocols stack is based on generally available protocols (see Figure 30). Testing must be performed with different types of solutions. It is quite hard to develop and add new lower-level SW parts (IP >>> IPX). There are no driver development kits.

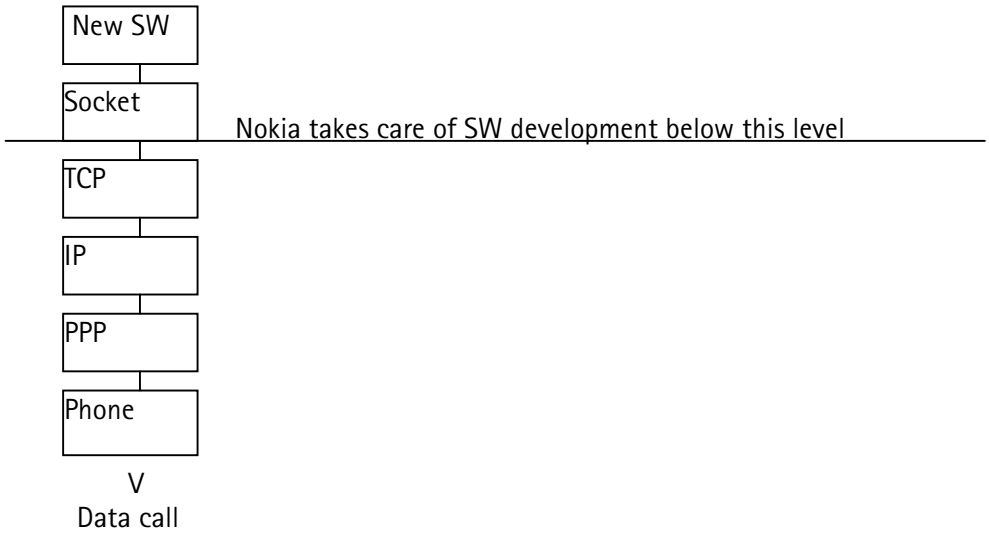


Figure 30. Internet communication protocol layers.

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

More information about establishing a PPP-connection can be found at Forum Nokia [44].

9.5 Taskswitching in the Internet application

Generally there can be only one instance of an Internet application running. However, there are some transitions when switching between applications:

- If the Internet application button is pressed for the first time after system boot, the Internet display is shown.
- If the Internet application button is pressed and no Internet applications are active, the Internet display is shown.
- If the Internet application button is pressed and at least one Internet application is active and the user is not currently using an Internet application, the view which was shown when the user left the Internet application is shown.

If the Internet application button is pressed and at least one Internet application is active and the user is currently using an Internet application, the Internet display is shown.

Some help for coding:

1. Opening a connection:
If SocketOpenDomainMedium returns SE_MEDIUM_BUSY, then the Nokia 9110 Communicator has another connection open which uses a different Internet Access Point. In this situation the following dialog should be shown:

Note:
You are connected to the Internet via IAP1. IAP2 is specified for this site. Use the active data call or make a new connection?

Use active
Make new Connection

Cancel

2. Closing a connection:
When you close a connection, the SocketCloseDomainMedium function should be given to force parameter value: FALSE, which means that the connection will be closed only if there is no other application using the connection.
If the SocektCloseDomainMedium returns something other than zero, then the connection was not closed, and the following note should be shown:

Note:
Cannot end data call because it is in use by another application.

OK

9.6 How an application should react in different error and informative situations

The following instructions concern all the applications, which establish data connections over the Socket API.

[44] http://www.forum.nokia.com (sdialin.pdf)

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

When Socket-specific errors occur

These error values are presented in your SDK [45] and [46]. Applications notice these errors from error return values from the Socket API. One example of a visible error note is:

"Internet connectivity problem: Temporary failure. Try again later."

The application should react to these error values differently depending on the value. As an example, if one of the following error values occurs, the application is not meant to give any error note:

- SSDE_DEVICE_ERROR:
- SSDE_NO_ANSWER:
- SSDE_LINE_BUSY:
- SSDE_NO_CARRIER:

These error values are high bytes, which are defined in the sockmisc.h file.

If an application receives a Socket error value like the above, it is not practical to inform the user about it. It is enough that the application notices it and tries, for example, the connection again later. For more information concerning opening and closing operation error values, contact the developer support group [47].

When making application notes, you should take into consideration that the user must be able to understand those notes, hence do not make explanations too technical. In conclusion, the principle is that there should not be both GSM- and Socket-specific error notes on the screen at the same time.

[45] pcgeos\N9110v10\Cinclude\socket.goh
[46] pcgeos\N9110v10\Cinclude\sockmisc.h
[47] 9110.sw.developer@nokia.com

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

10 CONTACTS

This chapter provides an overview of the Contacts application. Contacts is used for creating and deleting contact information. Created information is saved in the Contact DataBase, where it is available for other communication applications and the phone panel. The DataBase is designed for storing all kinds of contact information such as names, phone numbers and mail addresses.

There are two libraries (the ContDB and the ContLog [48]) which trace the Contact database and provide an API for the applications which have access to it:

- ContDB library
  - keeps track of the contact information
  - is used for editing and looking up the information
  - allows the selection of a contact from a list of existing items
- ContLog library
  - is used for logging what kind of communication has been made
  - allows one to choose recently contacted items

For more information about the Contact DataBase, please see the SDK [49]. There are also instructions for setting up a ContactList at the same pointer. A ContactList is used for presenting a list of known contacts.

10.1 Main view

The main view is used for creating new and opening existing contact entries (Figure 31).

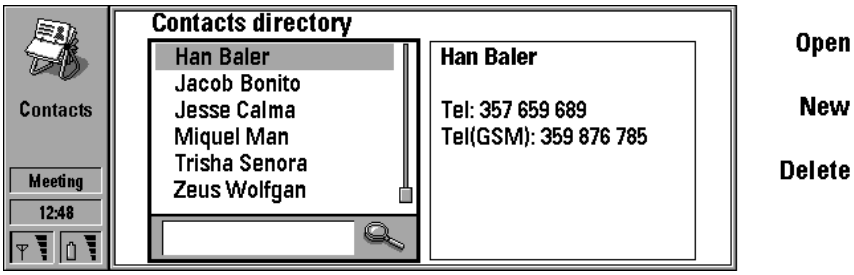


Figure 31. Contacts main view.

The name list can be scrolled with the arrow keys. Alternatively a desired contact can be found by using a search field. It is recommended to use the Search box if there is a long list of items. For more information, see Chapter 5.4.7 and the SDK [50]. There is more information about the Contact ASCII file format in the SDK [51].

Maximum character amounts for the fields are:

- Name 100,
- Company 100,
- Job Title 150,
- Address 150,
- Tel 30,
- Fax 30,
- Mail 1024,
- Dual Tone Multi Frequency 30

[48] pcgeos\N9110v10\Cinclude\contdb.goh and contlog.goh
[49] pcgeos\Techdocs\Nokia9000\ContactDB\index.htm
[50] pcgeos\N9110v10\Appl\Sdk_9000\Contact\contact.goh
[51] http://www.forum.nokia.com (Contact ASCII.pdf)

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

- Note 400.

10.2 Image in contact card

The Contact Database supports an image field, which allows the possibility to add an image to a contact card (Figure 32). The dimensions of an image that can be attached are 80 * 96 pixels (width * height). The same image is shown during an active call. In that case the image is shrunk to half its original size. Functions for image file setting (ContactGetImageFile and ContactSetImageFile) are provided by the ContDB library [52].

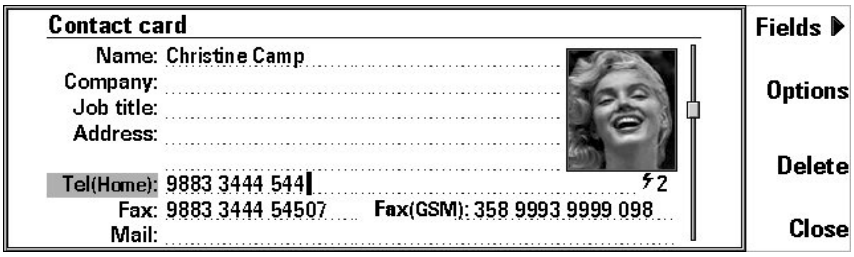


Figure 32. Attached image in contact card view.

10.3 Individual ringing tones

It is possible to choose a ringing tone for each existing contact. Selection is made in an opened contact card view by pressing Options. When an individual ringing tone is set, a note symbol is shown in the contact card. Functions for ringing tone field setting (ContactGetToneFile and ContactSetToneFile) are provided by the ContDB library.

[52] pcgeos\N9110v10\Cinclude\contdb.goh

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

11 NOTES

This chapter provides an overview of the UI of the Notes application. The Notes application is used for creating, editing, reading, printing, deleting, copying and renaming documents. All documents inside the Nokia 9110 Communicator are found from the Notes application.

11.1 Notes main view

The Notes main view shows the folders of the different document groups and the Document outbox (Figure 33).

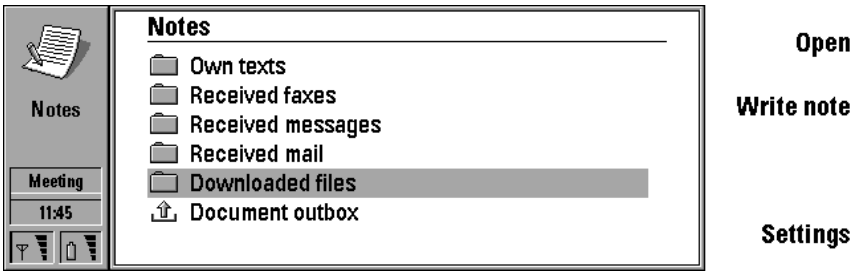


Figure 33. Notes main view.

11.2 Note editor

In the note editor it is possible to enter and format text (see Figure 34). The document name is shown in the editor header. When a document is opened, the name is placed in the header.

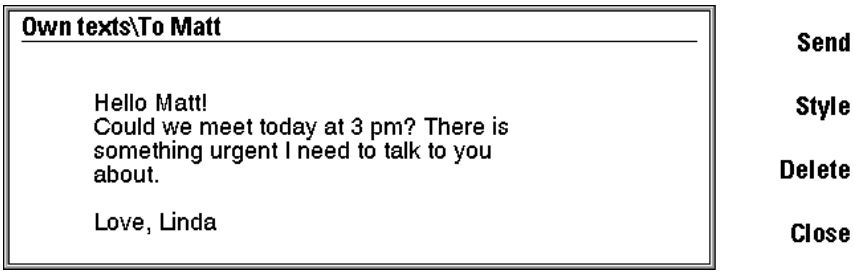


Figure 34. Note editor.

Text can be modified using the style list. The most commonly used text formatting tools include boldface, italics and underlining. When the style button is pressed, the Note editor style view is opened.

11.3 Received faxes, Downloaded pictures and DOS files

If a received fax is opened, the Fax viewer is activated. The received faxes are read only documents. For example WWW downloaded documents or Captured files are saved in the Downloaded files folder. When a Downloaded picture is opened, the Image viewer is activated. Pictures can be printed and sent as a fax. If a DOS file is transferred via PC connectivity to the Nokia 9110 Communicator, and opened in Notes, it is displayed as read only in the Note editor. The document also can be printed and sent as a fax.

11.4 Send and Print functionalities

It is possible to send any document to any media (note the restrictions with SMS and Mail). Sending options are by Fax, Short Message or Mail.

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

Documents can be printed only from the Notes application. After opening a document, press Menu button -> Print (see Figure 35). The printing interface is provided by the PrintControlClass, which can be found in the Spool library [53]. The documentation for the library is in the SDK [54].

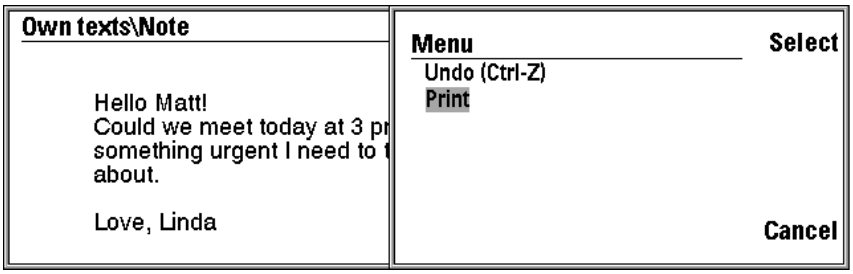


Figure 35. Printing a note.

[53] pcgeos\N9110v10\Cinclude\spool.goh
[54] pcgeos\Techdocs\Communication\Printing\index.htm

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

12 CALENDAR

This chapter provides an overview of the UI of Calendar application. The Calendar is an application including Month, Week and Day views. In the Month view the user can move from month to month and select any special day to mark or to view events. In Week view the user can see reserved times graphically for the whole week and move from week to week. In the Day view the user can view or change events and also choose the Details of the event. In the Details view the user can mark alarms, book events in a remote calendar, make memos or set repeating events.

There is more information about Calendar ASCII file format at Forum Nokia [55]. An API exists for adding appointments and to-do list items and for handling the added events (the API is defined in calendar.goh). More information about the Calendar API and also some sample applications can be found in the SDK [56].

12.1 Month view

Currently selected days that have events are indicated in Month view (Figure 36).

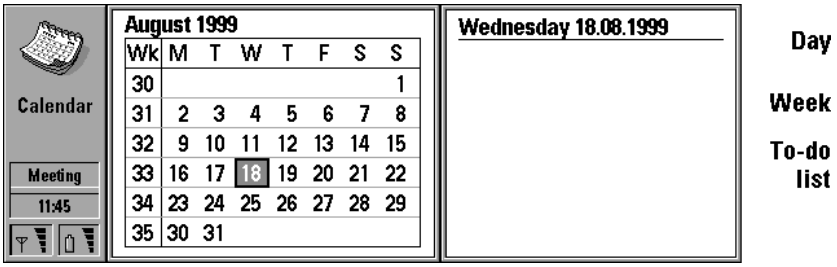


Figure 36. Calendar month view.

With the help of the Menu button it is easy to move, for example, between different months or weeks.

12.2 Week view

In Week view, reserved times are shown for the whole week (Figure 37). The bars are marked in grey, while the reserved whole day event bar is dark grey.

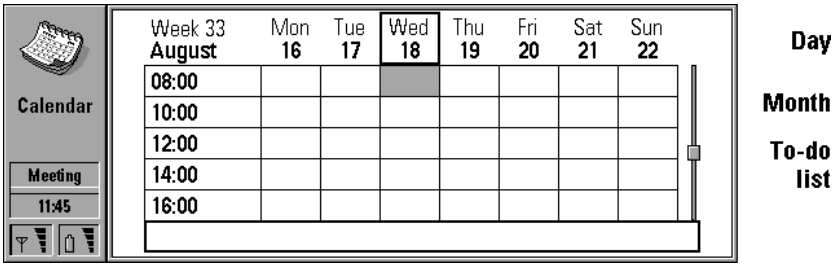


Figure 37. Calendar Week view.

12.3 Day view

Day view is for viewing events that have been written earlier and for adding new events (Figure 38).

[55] http://www.forum.nokia.com (Calendar ASCII.pdf)
[56] pcgeos\Techdocs\Nokia9000\CalendarAPI\index.htm

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

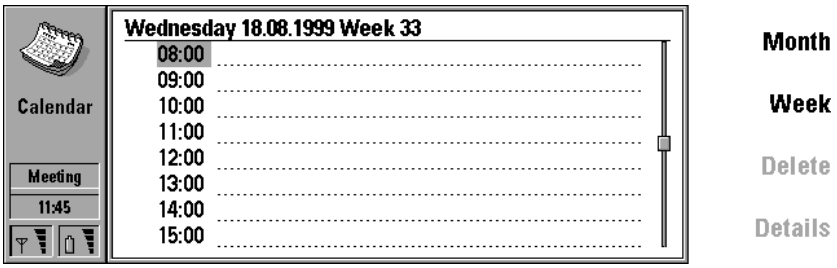


Figure 38. Calendar day view.

12.3.1 Details and Memo

Pressing the Details button opens an event-related Details view (Figure 39). In Details the user can change event Start/End Dates and Times, Alarm and Frequency. The user can reserve whole days by entering the number of days.

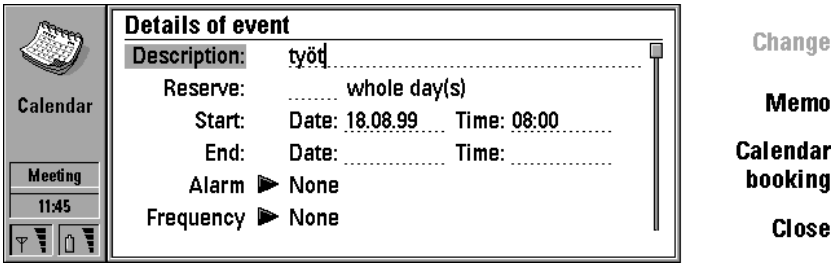


Figure 39. Details view.

The user can set an Alarm for the selected event. When the alarm has occurred, a note pops up on the screen. The default frequency is None and other available options for event frequencies are Daily/ Weekly/ Monthly/ Annually/Every other week and Working days.

Pressing the Memo button opens a memo view. It is the same as the Notes editor. If a memo is deleted from the Notes application, it is also removed from the Calendar.

12.3.2 Calendar booking

Pressing the Calendar booking button in the Details of event view opens the Booking view (Figure 40). The Booking functionality allows the user to send event requests and reservations to remote Calendars and to change and delete sent requests and reservations. The event will be sent as an SMS message or transferred via IR.

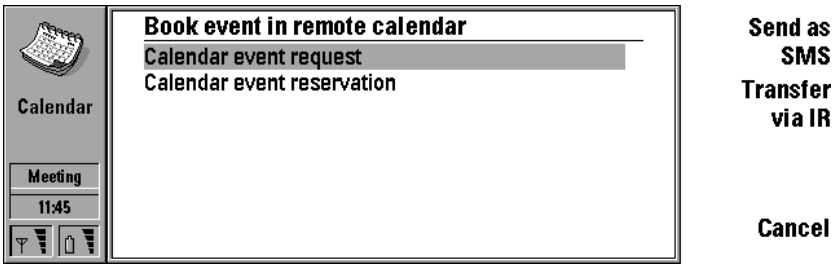


Figure 40. Calendar booking view.

When you have made the Calendar event reservation and selected the recipient, the application asks for the booking password. The booking password can be changed and it is given to users who allow others to send event reservations to their Calendars. The booking password is saved in System Security settings and the maximum length is 8 characters. It is possible to add a password to the Contact card.

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

13 SYSTEM

The System application is not actually an application in itself, but merely a place from which to launch System-related functions. Functions are launched from the System main view, which is shown in Figure 41.

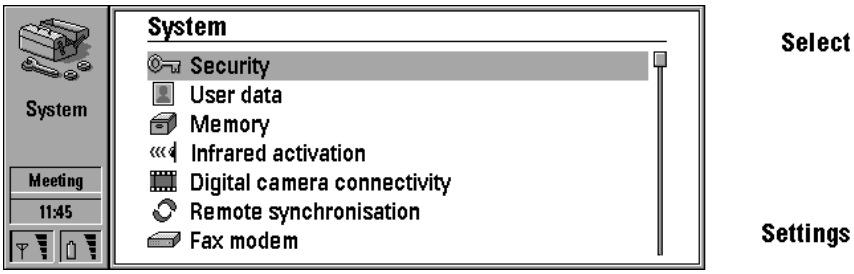


Figure 41. System application main view.

The System application contains functions, such as:
- the item for handling security issues,
- the item for installing/removing applications to/from the system,
- the item handling desktop connectivity to maintain data inside the device,
- the item for removing different data groups from the system,
- the item for importing/exporting contact or calendar information from/to the system.

13.1 Security

The Security item is placed under the system application. Security includes all security-related functionality and settings (see Figure 42).

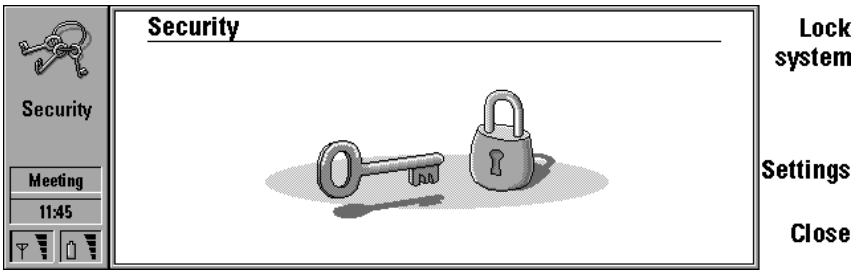


Figure 42. Security main view.

13.1.1 How to make a Serial number inquiry program

It is possible to use an IMEI code-based solution to make a Serial number inquiry. Please read Chapter 20, which introduces the IMEI code. There is also a code example in the SDK [57].

13.2 User data

The User data record is a place where user-specific information such as name, address, etc. is edited and saved. The basic functionality of the User data view is quite similar to the Contacts card view. The user has to fill in the fields on the contact card. The user sets these parameters when the Nokia 9110 Communicator is powered on for the first time. There is a URL field, where it is possible to add the user's URL address. This information is used by:

- the Fax application in the cover page "from information" and
- SMS "send business card" functionality.

[57] pcgeos\Techdocs\Nokia9000\Versions\Update_3.htm

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

13.3 PC Connectivity

The text, calendar and contacts data in the 9110 are Windows compatible. There are three methods of ipdating PC communicator files:

- via infrared (IrDA)
- by cable
- by cable with the optional desktop stand.

PC Suite for the Nokia 9110 Communicator is supplied on the CD-ROM, which comes in the sales package. With PC Suite the user can for example synchronise data, transfer files and install/remove software. Data can easily be backed up/returned and imported/exported with PC Suite. Figure 43 illustrates how the Nokia 9110 is connected with to a PC by a serial cable. For more detailed information, please see Forum Nokia [58] and the Nokia 9110 User guide.

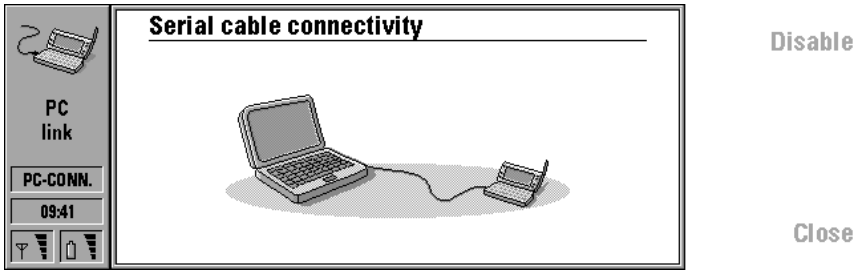


Figure 43. Serial cable connectivity view.

13.4 Wireless Imaging

It is possible to transfer images between the communicator and a digital still camera or another communicator. The transmission happens via the infrared connection and is enabled by the Digital camera connectivity application. To be compatible with the Nokia 9110 Communicator, a digital camera has to support the "IrTran-P" infrared protocol. The imported images are saved in the Downloaded files folder.

13.4.1 File formats and sizes

The Nokia 9110 Communicator supports JPEG and UPF image files. When sending images from the Nokia 9110 via the Internet, JPEG file format is used. UPF file format is used when transferring images between the Nokia 9110 and other devices, for example digital cameras. The recommended file size maximum is approximately 300 KB. When sending images there has to be at least 100 KB more free memory in the Nokia 9110 Communicator than the size of the image to be sent.

13.4.2 Menu view

Pressing the Menu button in the Digital camera connectivity main view enables you to:

- Open, Delete or Convert an image to JPEG
- See information about the image

13.5 Infrared receiving and remote synchronisation

The infrared receiving application is needed, when you want to use the PC Suite for the Nokia 9110 with the infrared connection. This application is also required to receive data from another device via the infrared connection.

[58] http://www.forum.nokia.com (Search: PC Suite)

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

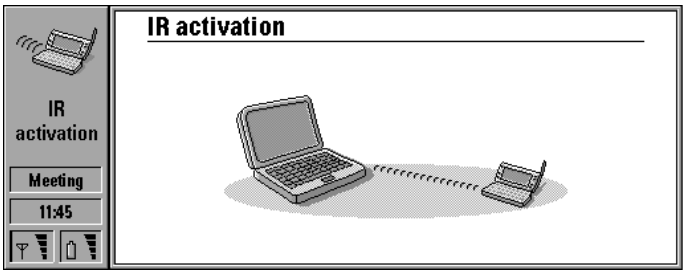


Figure 44. Infrared activation.

The Remote synchronisation application is needed, when you want to synchronise calendar data with your PC calendar. The PC Suite for the Nokia 9110 has to be running on the PC. The connection is made via the Internet so you will need an Internet Access Point (see Figure 44).

13.6 Memory issues

13.6.1 Viewing the memory

The memory option can be selected in the System main view. Opening the view shows the amount of free memory. The Details button opens the Memory details view, which is illustrated in Figure 45. Memory view comes with the flashing note "Please wait," until calculations are done and memory details are performed in the Memory details view.

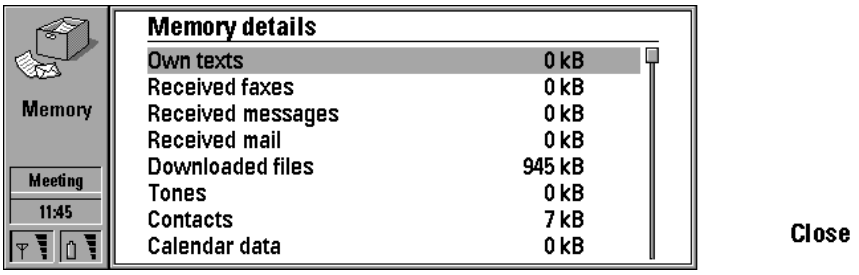


Figure 45. Memory details view.

Items in view are: Own texts, Received faxes, Received messages, Received mail, Downloaded files, Tones, Contacts, Calendar data, Installed applications and Temporary files. The view is scrollable and updated when it is opened. Temporary files are normally saved in the temporary files directory. When you are writing an application, which requires use of memory, you need to know how much memory is available on the 9110´s flash file system (FFS) [59].



13.6.2 Memory Card

The communicator memory can be expanded with a 4 MB or 8 MB memory card. Information about the Memory card tool is found in the System main view. By selecting this option the user is able to format, rename and check the card.

The Memory card is related to many applications in the Nokia 9110:

- The Memory card option is found in the Fax, SMS and Notes main view, which means that these items (faxes, short messages and notes) can be stored on the memory card.

[59] pcgeos\Techdocs\Nokia9000\MemLevels\index.htm

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

- The total and free memory of the card can be checked by selecting Memory -> Memory card in the System main view.
- It is possible to backup 9110 data to the memory card by selecting Backup/Restore in the System application.
- The contacts database can be created on the memory card by pressing Menu button -> New contacts database in the Contacts main view.
- Software is installed on the memory card using the Install/Remove application.

When a user starts the Nokia 9110 with the memory card, Geos merges the directories on the card with the corresponding 9110 directories. More information about the memory card and istructions how to install it, is found in the User guide.

There exists a special API which allows a developer deal with files and directories on a memory card. When developing geodes that use the memory card, this API should be known. The API can be found in the SDK [60]. The memory card can be emulated on the emulator using a floppy disk. Information about emulating is located in the SDK [61].

13.6.3 Low memory

In low memory situations there are two levels:

- Warning level (WL)
- Critical level (CL)

The storage space of the device is called the Flash File System (FFS). The warning level is a memory limit where the user gets a warning about low memory. The warning level is 50 KB. The Critical level is a memory limit where saving to FFS is no longer possible. If this limit is passed the software state cannot be saved and data may be corrupted. The critical level is 25 KB.

The notes are:
1. "Insufficient memory to perform operation. Delete some documents or move some documents to memory card or PC and try again." {OK}
2. "Not enough memory to execute operation. Delete some documents and try again." {OK}
3. "Insufficient memory to save changes. Close will discard changes. Cancel allows you to go to another application and delete some documents. Return to Close this document after that.{OK}
4. "Running out of memory. Delete some documents." {OK}
5. "Memory full. Delete some documents." {OK}

If WL is exceeded note 4 is displayed. However there are some situations when the WL is exceeded but no note is displayed, such as:
- state is saved,
- contact log is updated,
- writing to recent contacts stack,
- Copying to clipboard.

Note 4 is displayed when CL<memory≤WL and when
- the system is booted (battery off),
- the cover is opened,
- the system is unlocked.

[60] pcgeos\Techdocs\Nokia9000\MMC\api.htm
[61] pcgeos\Techdocs\Nokia9000\MMC\emulate.htm

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

Note 5 is displayed when memory≤CL and when
- the system is booted,
- the cover is opened,
- the system is unlocked.

Note 3 is displayed when the user has edited an editable view and CL is exceeded.

Note 2 appears in other cases when the CL has been exceeded. If the CL is exceeded at the start or during an operation, the operation is terminated and:
- a partially received fax is saved,
- partially received mail is not saved,
- reading/fetching mail is cancelled and the file deleted,
- downloaded WWW files or imported contact files or an installed application or captured text or a copied document are deleted,
- printing is cancelled.

Note 1 appears when memory is between WL and CL and any editable view is opened or any memory-intensive operation is started.

13.7 How to make a Software version inquiry program

You can verify the software version in the System application "About this product." The version number can be found at the bottom of the scrollable view (see Figure 46).

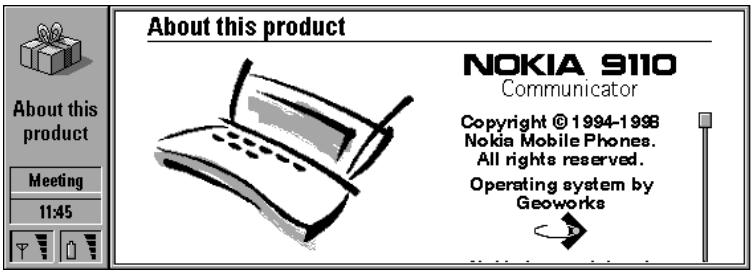


Figure 46. About this product view.

The version number of the software is stored in the .ini file. The Category is UI and the key is swSerialNumber. The version string is 27 characters long. This string can be seen on the last line of "About this product". The string has the following format: 000000/00/230040801044201/0. The important part is: 230040801044201, where
- the three first characters identify the hardware version (here 2.30)
- the following six characters identify the software version (here 4.8.1)
- the following three characters identify the country code (here 044 = English)
- the next character is 2
- and the last two characters identify the keyboard used (here 01 = UK).

A code example is in the SDK [62]. For different language versions, see Chapter 18.2.

[62] pcgeos\Techdocs\Nokia9000\Versions\Update_1.htm

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

14 EXTRAS

The Extras application is not actually an application in itself, but it is a place to launch applications that are not classified under any other application button. When the Extras button is pressed, the previously activated application under the Extras list is activated again. By pressing the Extras button another time, the Extras list is displayed again. By pressing the Close command button, the Extras main view is displayed again.

When the Extras list is activated, the main view is opened on the screen (see Figure 47).

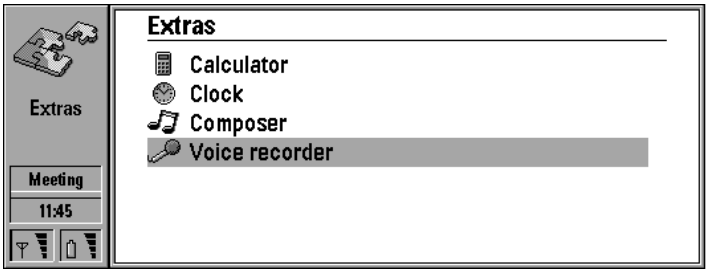


Figure 47. Extras view.

Applications can be selected by using the arrow keys. More applications can be added to the Extras list by using the application installation feature. There is more information about how to distribute an application and how to make an installation file in Chapter 21.

14.1 Voice recorder

Telephone conversations and one´s own speech can be recorded with the voice recorder. The maximum length of a recording is 30 seconds. Own recordings and other sound files can also be listened to and set to ringing tones. The recommendation for file format is an 8 bits mono .wav file when recording from a PC to the 9110. The file format from the 9110 to a PC should be RT24 coded. Recording is possible using the "vrecui library". Functions for recording voice and playing .wav files are found in the SDK [63].

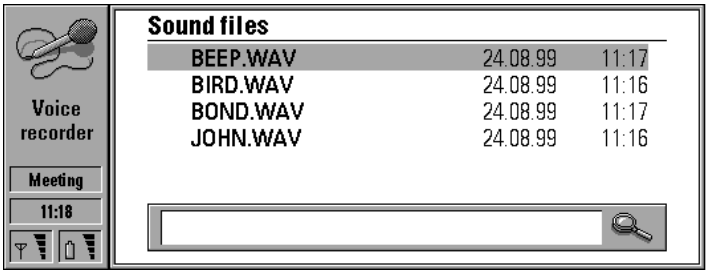


Figure 48. Sound files directory.

14.2 Taskswitching in the Extras application

It is possible to start more than one of the applications in the Extras list. The following rules are followed:

- If the Extras button is pressed when one of the applications in the extras list is already active on the screen, the Extras list is activated.
- If the Extras button is pressed when some other application is active on the screen, the last of the previously activated applications of the Extras list is activated again.
- Every application launched in the Extras list maintains its state.
- It is not possible to launch the same application in the Extras list twice.

[63] pcgeos\N9110v10\CInclude\Internal\Resp\vrecui.goh

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

15 HELP TEXTS

This chapter provides an overview of the UI of the help application. The purpose of the Help application is to provide quick help to the user for the currently active screen. It is recommended that help be available for all applications. The user can also get underlying application help topics and general help for the Nokia 9110 Communicator.

15.1 Help view

Pressing the Help key (i) causes a screen-sensitive Help view to appear over the currently running application. An example is shown in Figure 48. If the Help button is pressed in a view, the current application icon and name are displayed on the left side of the screen.

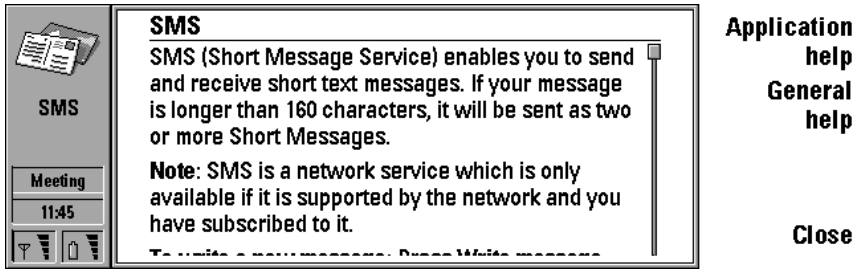


Figure 49. The Help view in Contacts.

If the text is too long to fit on the screen, it can be scrolled. If there is not a special help view for a currently active application, the Application help screen is opened (see Figure 49). The General help view includes a scrollable list of all the common Help topics.

15.1.1 Application help

The Application help view includes a scrollable list. The list contains all the Help topics related to the underlying application. Application topics do not have to be in alphabetical order.

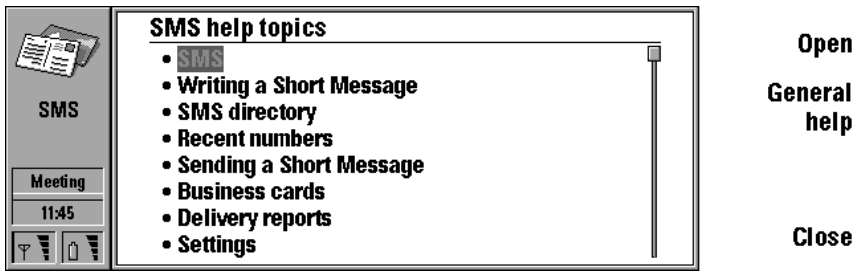


Figure 50. The Application topics view.

From the Application topics the user can move to General topics view and back again.

15.1.2 General help

The user can select topics the same way as in Application topics. Topics in the General topics view can also be moved to the Application topics view.

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

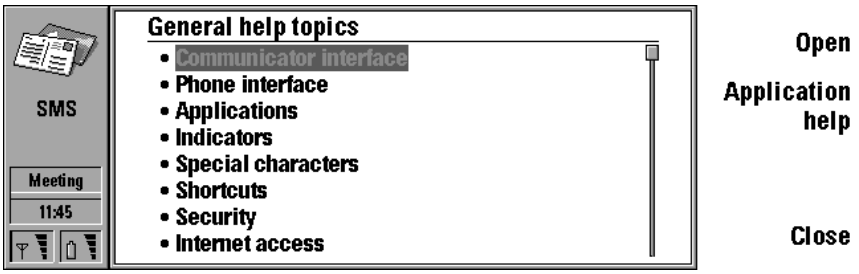


Figure 51. General topics view.

15.2 How to make help files

It is best to use Geowrite for making help text for an application. You can find Geowrite in the SDK by turning on target (geos2xnc) [64].

15.2.1 Enabling the help editor

1. Make changes to geos.ini file

[configure]
helpEditor = true.

2. Start GeoWrite.
3. Press Change User Level and select level 4.
4. Press Fine Tune and enable the Help Editor by clicking the radio button.
5. Open or create a file in GeoWrite.
6. Check that Help Editor is in the menu.
7. Select Options, Save Configuration.

15.2.2 Writing help files in GeoWrite

1. The help file name must be exactly 8 characters long, spaces are valid.
2. Each separate help screen must be on its own page, press Ctrl-Enter to make page breaks.
3. Write the help pages.
4. Write a table of contents page for the file.
5. Remember to save your work regularly.
6. Help file DOS names automatically have an extension ".000".

15.2.3 Setting context points and hyperlinks

1. The application finds the correct help screen by referring to the help file name and the help context point (each help page within a help file must have its unique context).
2. To define help contexts (one of these must always be "TOC") in the Help Editor, select Help Editor, Define Context.
3. To set the help context for a page, first select the title of the page, then select Help Editor, Set Context, select the context and select Apply.
4. Set the help context for the table of contents page as "TOC" (this will be used when the user presses "Application topics" in help).
5. To set the hyperlinks in the table of contents page, first select the appropriate text, then select Help Editor, Set Hyperlink, select the context and select Apply.
6. To get rid of any extra/unnecessary bits of the link (at the end of a line etc.) or wrong/unnecessary links, select the area with Shift and arrow keys, then select Help Editor, Set Hyperlink, remove the link by double clicking the selected context and select Apply.
7. To set hyperlinks to other help files:
   * select Help Editor, define file

[64] pcgeos\Techdocs\Ui\Help\index.htm

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

* select Help Editor, define Context and define the context for the other file,
* select the appropriate text,
* select Help Editor,
* set Hyperlink,
* select the context from the other file and
* select Apply.

15.2.4 Generating help files

1. The applications use "generated" versions of the help files, keep them separate from the editable GeoWrite files (the names are identical).
2. To generate help files, select Help Editor and Generate help file (make sure that the Compress Data in the File option is on).
3. The generated help files go to (userdata\help).
4. The generated help file DOS names automatically have a .000 extension.

15.3 Help style guide

Good things to keep in mind when developing Help screens for an application.

1. Remember that the user looks at the help, when he:
    - is stuck OR
    - wants to know something specific about the system.

Therefore, keep your Help application simple and clear

2. Try to imagine what the user wants to find out from the help screen.

Usually the situation is either A or B. For example if the question is:

- "How do you do 'X'?"...to do 'X', press Z.
- "What does 'Z'- do?...Z enables you to do 'X'.

Decide which is the most likely question construct and answer appropriately.

3. Avoid the passive if possible.

However, you can use such constructions as:

- can be sent...
- is stored...

4. Follow a logical order.

For example:

- explain a sequence of steps sush as "To do 'X', open 'Z' then select 'B', then press 'C'.
- explain the command buttons in the order they appear on the screen, e.g. 'U' does... 'V' allows you to... 'W' enables you to...

5. Stick to conventions.

Command buttons should be in bold.

6. Avoid repetition.

Actions and commands to the whole system do not need to be explained everywhere.

7. Explain new and unusual features or terms when necessary, for example DTMF tones.

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

16 INDICATORS

This section describes the interface used to control the Indicator application. Generally, space should be left for the Indicator area, but in special cases it is possible to use the entire width of the screen.

The Indicator area is placed in the lower left corner of the screen (see Figure 52). The Indicator area requires 90 * 80 pixels (height * width). The rest of the height (110 pixels) is used by the application identifier icon and application name. The size of the application icon is 64 * 50 pixels. The name of the application is written in Systemfont 16 bold. Normally, the name fits on one line, but if the name must be on two lines, use \r in the middle of the name (for example Developer\rApp). Not all indicators are presented in this section.

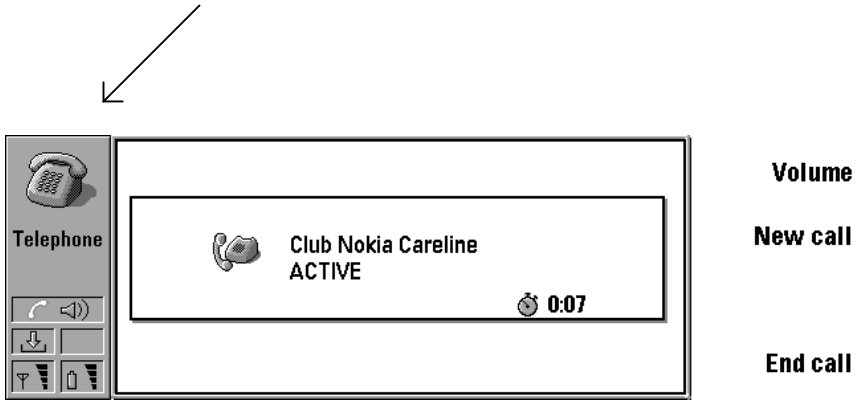


Figure 52. Indicator area on the screen.

The Indicator area is divided into four sub-areas:

- Active connection state / Profile
- Inbox / Outbox
- Field strength indicator
- Battery charge level

The PC connection, printing and Fax-Modem mode cannot operate at the same time. That is why they are indicated in the same sub-area, the Active connection\Profile area. Also Active Fax/Data/Voice call and Profile indicators are shown in this area. These indicators are (see Table 3):

- PC connection (Indicator 1)
- Print (Indicator 2)
- Fax-Modem mode (Indicator 3)

- Fax being sent or received (Indicator 4)
- Data call is leaving or received (Indicator 5)
- In Voice call the personal hands free is on (Indicator 6)
- In Voice call the personal hands free is off (Indicator 7)
- Profile indicator (Silent) (Indicator 11)

Infrared indicators are shown in the top most area. If a slot is used for active call indication, these indicators are not shown. These indicators are:

- Status OK or connected (Indicator 8)
- Status blocked (Indicator 9)
- Status disconnected (Indicator 10)

Other Indicators in Table 3 are:

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

Time	(Indicator 12)
Unread message in inbox	(Indicator 13)
Document in outbox	(Indicator 14)
Field	(Indicator 15)
Cellular network is unavailable	(Indicator 16)
The phone interface is switched off	(Indicator 17)
Battery low, the phone will be turned off	(Indicator 18)
Battery	(Indicator 19)
Charging	(indicator 20)

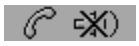
Indicator number	Indicator	Indicator number	Indicator
Indicator 1.		Indicator 11.	
Indicator 2.		Indicator 12.	
Indicator 3.		Indicator 13.	
Indicator 4.		Indicator 14.	
Indicator 5.		Indicator 15.	
Indicator 6.		Indicator 16.	
Indicator 7.		Indicator 17.	
Indicator 8.		Indicator 18.	
Indicator 9.		Indicator 19.	
Indicator 10.		Indicator 20.	

Table 3. Indicators.

16.1 Indicator application interface

The software interface is built using the General Change Notification List mechanism of GEOS operating system and SysSendNotification mechanism.

The indicator application is registered to the GCN list GCNSLT_NOTIFY_INDICATOR_EVENT and the same notification list via the SysHookNotification mechanism . Notification messages used are MSG_META_NOTIFY and MSG_META_NOTIFY_WITH_DATA_BLOCK. The manufacturer identifier used in the GCN list is always MANUFACTURER_ID_GEOWORKS.
For more information, please see the SDK [65].

[65] pcgeos\N9110v10\CInclude\geoworks.h

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

16.2 How to set an application icon

Sets the application icon displayed in the indicator window.
(GCN)Message:

- MSG_META_NOTIFY_WITH_DATA_BLOCK

Notification type:

- GWNT_INDICATOR_REPLACE_ICON_GLYPH.

Parameters:

- MemHandle data (IndicatorDataBlock):

* IndicatorDataBlock.monikerSize (size of the moniker in bytes, 0 = no application icon).
* IndicatorDataBlock.monikerData.iconMoniker (icon moniker).

16.2.1 How to make your own application identifier icon

One example of such code:

```
@object GenApplicationClass MyApplicationObj = {
   GI_visMoniker = list { @APPLICATION_TITLE, @APPLICATION_ICON }
   GI_comp =
   ...
   }

@visMoniker  APPLICATION_TITLE = "My application";
   /* To separate the text to two lines use /r */
@visMoniker  APPLICATION_ICON  = {
        size = standard;
        style = icon;
        colour = color4;
        aspectRatio = normal;
        cachedSize = 64, 50;
        gstring {
                        GSDrawBitmapAtCP (1013),
                        Bitmap (64,50,BMC_PACKBITS,(BMT_MASK|BMF_4BIT)),
                        0xf9, 0xff, 0xe1, 0xdd,
                        ...
                        0xf9, 0xff, 0xe1, 0xdd,
                        0xf9, 0xff, 0xe1, 0xdd,
                        0xf9, 0xff, 0xe1, 0xdd,
                        GSEndString()
        }
   };
```

It is recommended to use Icon Editor in the SDK using a maximum of 16 colours.
For more information, please see [66].

16.3 Setting application text

Sets the application text displayed in the indicator window.
Message:

- MSG_META_NOTIFY_WITH_DATA_BLOCK

[66] Geoworks Nokia 9000 Communicator Programming Class Notes\Nokia 9000 Communicator Complex Monikers

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

Notification type:

- GWNT_INDICATOR_REPLACE_TEXT_GLYPH.

Parameters:

- MemHandle data (IndicatorDataBlock):

* IndicatorDataBlock.monikerSize (length of the moniker in bytes, 0 = no application text).
* IndicatorDataBlock.monikerData.iconMoniker (text moniker)

16.4 Setting the number of unread received documents

Sets the number of unread documents of a given type.
Message:

- MSG_META_NOTIFY_WITH_DATA_BLOCK

Notification type:

- GWNT_INDICATOR_SET_NUM_UNREAD_DOCUMENTS

Parameters:

- MemHandle data (IndicatorDocInfo):

* IndicatorDocInfo.geodeToken (the geode token that uniquely identifies document type).
* IndicatorDocInfo.singularText (text to display in the lid open window if the number of documents is one).
* IndicatorDocInfo.pluralText (text to display in the lid open window if the number of documents is greater than one).
* IndicatorDocInfo.docCount (number of unread documents).

16.5 Changing the number of unread received documents

Changes the number of unread documents of a given type. The inbox indicator will be set accordingly.
Message:

- MSG_META_NOTIFY_WITH_DATA_BLOCK

Notification type:

- GWNT_INDICATOR_CHANGE_NUM_UNREAD_DOCUMENTS

Parameters:

- MemHandle data (IndicatorDocCountInfo):

* IndicatorDocCountInfo.geodeToken (the geode token that uniquely identifies document type).
* IndicatorDocCountInfo.docCount (if docCount is negative, the number of unread documents decrements by one (-1). If docCount is positive, the number of unread documents is incremented by one (+1).

16.6 Setting the number of outgoing documents

Sets the number of outgoing documents.
Message:

- MSG_META_NOTIFY

Notification type:

- GWNT_INDICATOR_SET_NUM_OUTGOING_DOCUMENTS

Parameters:

- Word data (number of outgoing documents)

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

16.7 Changing the number of outgoing documents

Changes the number of outgoing documents.
Message:
- MSG_META_NOTIFY

Notification type:
- GWNT_INDICATOR_CHANGE_NUM_OUTGOING_DOCUMENTS

Parameters:
- Word data (set to non-zero, if number of outgoing documents is to be incremented by one; set to zero, if number of outgoing documents is to be decremented by one)

16.8 Setting custom notes

Displays/removes custom text or icon note in the uppermost area of the indicator window.
Message:
- MSG_META_NOTIFY_WITH_DATA_BLOCK

Notification type:
- GWNT_INDICATOR_SET_CUSTOM_NOTE

Parameters:
- MemHandle data (IndicatorCustomNote):

* IndicatorCustomNote.geodeToken (the geode token that uniquely identifies text note).
* IndicatorCustomNote.monikerSize (length of the moniker in bytes; set to 0 to have no custom note).
* IndicatorCustomNote.monikerData.iconMoniker (moniker).

16.9 Setting visibility of main window

Sets the visibility of the main window of the indicator application. The most important data from the indicator window is displayed even if the main window is turned off:
- call status information/custom note,
- battery low.

These indicators appear only when they would also appear in the normal indicator window. When these indicators are on, the indicator icons will blink.

Message:
- MSG_META_NOTIFY

Notification type:
- GWNT_INDICATOR_SET_MAIN_WINDOW_VISIBILITY

Parameters:
- Word data (set to non-zero to turn the main window on; set to zero to turn the main window off).

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

17. DOCUMENT OUTBOX

This chapter specifies the UI of the Document outbox.

All the documents that the user sends go through the Document outbox. It sends the documents whenever possible.

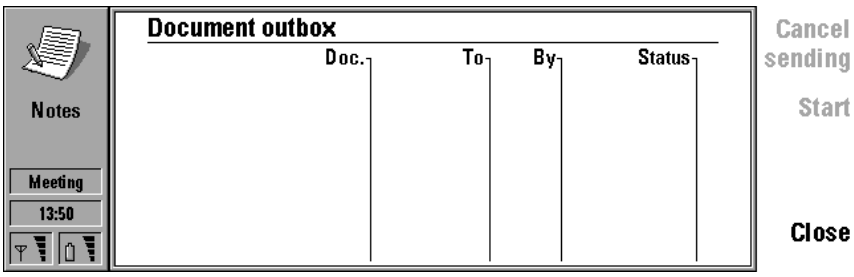


Figure 53. Document outbox view.

The Document outbox can be accessed in the Fax, SMS, Mail and Notes applications. To open the Document outbox, select the Document outbox from the list and press Open. As a special feature, a user can set a document to be sent out from the Outbox only when requested.

In Figure 53 the application view is divided into four different parts.

1. The Doc. field contains the name of the document to be sent.
2. The To field contains the receiver of the document (name or number or mail address).
3. The By field contains the transport type used in sending (value can be "Fax" or "Msg." or "Mail").
4. The Status field indicates the status of the document in the queue. Status value can be:
   "Waiting" (for the media to become available so that sending becomes possible),
   "Retry..." (documents that failed to be sent and are waiting to be placed in the queue),
   "Upon request" (mail documents will be sent only on user request),
   "Preparing" (documents are under preparation for sending),
   "Ready" (documents that are prepared and ready to be sent when their turn comes),
   "Sending" (documents that are currently being sent).

For an example of message sending through the outbox in the SDK [67].

[67] pcgeos\Techdocs\Communication\Mailbox\Mailbox16_htm

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

18. LOCALIZATION

Localization is the process of adapting software for a specific region or language. There must be different language versions of software available, because Nokia the 9110 is sold and used in several countries. It is recommended to get familiar with some principles about localization, if planning to release software in many languages. Documentation about this can be found in the SDK [68].

18.1 Resource Editor (ResEdit) Tool

The idea when using the ResEdit tool is to put localized text and bitmaps in localizable resources. ResEdit looks for localizable objects only in non-code resources, where the string or bitmap is stored in a chunk. When using this tool, the translator doesn´t need to see the source code or even know how to program. Information about the tool can be found in the SDK [69].

18.2 Language versions

There are 10 different UI versions:

Danish	045
English	044
Finnish	358
French	033
German	049
Italian	039
Norwegian	047
Portuguese	351
Spanish	034
Swedish	046

And six different keyboard mapping versions:

French	02
German	03
Italian	06
Scandinavian	04
Spanish/Portuguese	05
UK	01

18.3 File names

It must be noticed that the file names vary according to different language versions. This is extremely important from the developer´s point of view. All language versions should have their own .ins file, especially if there are some extra data files, that are not inside the application files. Other localized file names are shown in Table 4:

Names	DOS-name	Names	DOS-name
English		**German**	
Downloaded files	DOWNLOAD.000	Empfangene dateien	EMPFANGE.003
Own texts	OWN_TEXT.000	Eigene Texte	EIGENE_T.000
Received faxes	RECEIVED.000	Empfangene Faxmitteilungen	EMPFANGE.000
Received SMS	RECEIVED.001	Empfangene Kurzmitteilungen	EMPFANGE.001
Received mail	RECEIVED.002	Empfangene E-Mail	EMPFANGE.002
Document outbox		Ausgangskorb	
Standard messages	STANDARD.000	Kurzmitteilungsvorlagen	KURZMITT.000
Business cards		Visitenkarten	
Info messages	INFO_MES.000	Nachrichten	NACHRICH.000
Delivery reports		Übertragungsberichte	

[68] pcgeos\Techdocs\Programming\Localization\index.htm
[69] pcgeos\Techdocs\Tools\ResEd\index.htm

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

Remote mailbox		E-Mail-Server	
Tones	TONES.000	Klänge	KLÄNGE.000
Contact images		Bilder zu Kontakteinträgen	

Finnish		**French**	
Noudetut tiedostot	NOUDETUT.000	Fichiers téléchargés	FICHIERS.000
Omat tekstit	OMAT_TEK.000	Textes personnels	TEXTES_P.000
Vastaanotetut faksit	VASTAANO.000	Fax reçus	FAX_REÇU.000
Vastaanotetut tekstiviestit	VASTAANO.001	Messages reçus	MESSAGES.000
Vastaanotetut -		Courrier reçu	COURRIER.000
Sähköpostiviestit	VASTAANO.002		
Lähtevät viestit		à envoyer	
Vakioviestit	VAKIOVIE.000	Messages standard	MESSAGES.001
Käyntikortit		Cartes de visite	
Tiedotteet	TIEDOTTE.000	Infos par le réseau	INFOS_PA.000
Viestien välitystiedot		Accusés de réception	
Etäpostilokero		Boîte aux lettres distante	
Äänet	ÄÄNET.000	Audio	AUDIO.000
Osoitekirjan kuvat		Images fiches de contact	

Italian		**Spanish**	
File scaricati	FILE_SCA.000	Archivos recogidos	ARCHIVOS.000
Propri testi	PROPRI_T.000	Textos propios	'TEXTOS_P.000
Fax ricevuti	FAX_RICE.000	Faxes recibidos	FAXES_RE.000
Messaggi SMS ricevuti	MESSAGGI.000	Mensajes recibidos	MENSAJES.000
E-mail ricevuta	E-MAIL_R.000	Correo recibido	CORREO_R.000
Documenti uscenti		Bandeja de salida de documentos	
Messaggi standard	MESSAGGI.001	Mensajes estándar	MENSAJES.001
Biglietti da visita		Tarjetas de la Guía	
Messaggi info	MESSAGGI.002	Mensajes de difusión	MENSAJES.002
Rapporti ricezione		Informes de entrega	
Casella postale remota		Buzón remoto de correo	
Toni	TONI.000	Tonos	TONOS.000
Immagini Rubrica		Imágenes de la Guía	

Danish		**Swedish**	
Modtagede filer	MODTAGED.000	Nerladdade filer	NERLADDA.000
Egne tekster	EGNE_TEK.000	Egna dokument	EGNA_DOK.000
Indgående faxer	INDGÅEND.000	Mottagna fax	MOTTAGNA.000
Indgående beskeder	INDGÅEND.001	Mottagna textmeddelanden	MOTTAGNA.001
Indgående e-mail	INDGÅEND.002	Mottagen e-post	MOTTAGEN.000
Dokument udbakke		Utkorg	
Standard beskeder	STANDARD.000	Standardmeddelanden	STANDARD.000
Visitkort		Kontaktkort	
Serviceinfo meddelelse	SERVICEI.000	Infomeddelanden	INFOMEDD.000
Afsendelsesrapporter		Leveransrapporter	
Fjern-postkasse		E-postoperatör	
Toner	TONER.000	Toner	TONER.000
Kontaktbilleder		Kontaktbilder	

Copyright © 1999 Nokia Mobile Phones. All rights reserved.

Norwegian		**Portuguese**	
Nedlastede filer	NEDLASTE.000	Ficheiros importados	FICHEIRO.000
Egne tekster	EGNE_TEK.000	Textos próprios	TEXTOS_P.000
Mottatte fakser	MOTTATTE.000	Faxes recebidos	FAXES_RE.000
Mottatte tekstbeskjeder	MOTTATTE.001	SMC recebidas	SMC_RECE.00
Mottatt epost	MOTTATT_.000	E-mail recebido	E-MAIL_R.000
Utkurv		Caixa de saída de documentos	
Standardbeskjeder	STANDARD.000	Mensagens standard	MENSAGEN.000
Visittkort		Cartões de visita	
Nettverksinfo	NETTVERK.000	Mensagens de informação	MENSAGEN.001
Leveringsrapporter		Relatórios de entregas	
Epost-server		Caixa de correio remota	
Toner	TONER.000	Tons	MELODIAS.000
Kontaktbilder		Imagens de contactos	

Table 4. File names.

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

19. APIs

The following list introduces some API features that exist in the Nokia 9110 Communicator. For more information about these, please see the SDK [70] and [71].

Access Point library
- provides the interface for choosing and editing access point entries
- include: accpnt.goh
- documentation: pcgeos\Techdocs\Communication\AccPnt\index.htm

Calendar API
- provides the calendar handling interface
- include: calendar.goh
- documentation: pcgeos\Techdocs\Nokia9000\CalendarAPI\index.htm

ComplexMonikerClass
- provides command triggers and the Menu trigger interface
- documentation: pcgeos\Techdocs\Nokia9000\UI\9000UI_trigger.htm

ContDB library/ContdLog library
- provides Contact Database handling interface
- include: contdb.goh, contdb.h, foam.goh
- documentation: pcgeos\Techdocs\Nokia9000\ContactDB\index.htm

ContactListClass
- provides the contact list interface
- documentation: pcgeos\Techdocs\Nokia9000\ContactDB\ContactDB_1.htm

ContactList searches
- forces the ContactList to remember the current search
- pcgeos\Techdocs\Nokia9000\CotactDB\ContactDB_3.htm

ContactGetUserName();
- reads the user´s name
- include: contdb.h

EciGetImei();
- reads the phone´s IMEI number
- include: eci.goh

EciGetCellId();
- reads the Cell ID
- include: eci.goh

FilterListClass/FilterTextClass
- together provide the interactive searches interface
- include: foam.goh
- documentation: pcgeos\Techdocs\Nokia9000\UI\filter.htm

[70] pcgeos\Techdocs\Nokia9000\Versions\index.htm
[71] pcgeos\Techdocs\Nokia9000\UI\index.htm

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

GenItemGroupClass
- provides the non-exclusive lists interface
- documentation: pcgeos\Techdocs\Nokia9000\Versions\Update_6.htm, pcgeos\Techdocs\UI\Lists\Lists_8.htm

Mailbox API
- provides the interface for receiving and sending information
- include: malibox.goh, mailbox.def
- documentation: pcgeos\Techdocs\Communication\Mailbox\index.htm

MailboxSendControlClass
- provides the interface for setting a fax receiver
- include: mailbox.goh
- documentation: pcgeos\Techdocs\Communication\FaxSend\index.htm

Memory Card API
- allows dealing with files and directories on a card
- include: respondr.goh, foam.goh
- documentation: pcgeos\Techdocs\Nokia9000\MMC\api.htm

PCCom library
- provides the interface for communication with another device over a serial connection
- include: pccom.goh
- documentation: pcgeos\Techdocs\Communication\PCCOm\index.htm

PrintControlClass
- provides the printing interface and the interface for drawing a fax´s contents
- include: spool.goh
- documentation: pcgeos\Techdocs\Communication\Printing\index.htm, pcgeos\Techdocs\Communicaton\FaxSend\FaxsSend_2.htm

ProductName/Software serial number
- tells the software version and serial numbers
- documentation: pcgeos\Techdocs\Nokia9000\Versions\Update_1.htm

RespSetBacklightState (Boolean state);
- turns on/off the backlight
- include: respondr.goh

RespGetBacklightState (void);
- returns the current state of the backlight
- include: respondr.goh

RespGetPDAHardwareInfo (void);
- gets the PDA hardware info
- include: respondr.goh
- documentation: pcgeos\Techdocs\Nokia9000\Versions\Update_1.htm

RespGetPhoneModuleStatus (void);
- gets the current status of the phone
- include: respondr.goh

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

SMRegRegisterApplication();
- registers application to the SMS receive application
- include: smreg.goh
- documentation: pcgeos\Techdocs\Nokia9000\Versions\Update_4.htm

SMS Mailbox API
- provides the interface for short message sending
- include: mailbox.goh, foam.goh, string.h, vmtree.h, smdefine.goh, initfile.h
- documentation: pcgeos\Techdocs\Nokia9000\SMS\SMS_6.htm

Socket API
- provides the device´s internal and external communication interface
- include: socket.goh, sockmisc.h
- documentation: pcgeos\Techdocs\Communication\Socket.htm

StartPlayVoiceFile();
- starts playing voice
- include: vrecui.goh

StopPlayVoiceFile();
- stops playing voice
- include: vrecui.goh

StartRecVoiceFile();
- starts recording voice
- include: vrecui.goh

Zoom level API
- provides zoom feature interface
- include: foam.goh
- documentation: pcgeos\Techdocs\Nokia9000\UI\9000UI_zoom.htm

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

20. APPLICATION COPY PROTECTION

If you want to copy protect your application, one alternative is to use the IMEI code-based solution.

What is the IMEI code:

- a unique 15 digit code number which identifies a Nokia 9110 Communicator,
- users cannot change the code. The IMEI code is hidden in the hardware, not in the GEOS configuration files.

Advantage: It is possible to create reliable application software copy protection.

Reading the IMEI Code

The function takes one parameter, a pointer to a fill with the code number string. The buffer must be at least 17 bytes long to hold the null-terminated 16-character code string. The function returns zero if the IMEI code was read from the phone successfully; otherwise it will return one.

```
@include <eci.goh>

...

@method testProcessClass, MSG_TEST_PROCESS_GET_IMEI
{
        TCHAR imei[17] = " " ;

        If (EciGetImei (imei) == 0)
        {
         /* We got IMEI code, do any check you want. */
        }

        else
        {
         /* For some reason query failed. */
        }
}
```


Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

21. HOW TO DELIVER YOUR APPLICATION

The files that make up the software are known as the software package. The package includes all of those files, which are needed to use the product. No matter how the software is delivered, there is one file which must be provided: the Package Description File (.INS file). That is an ASCII text file that specifies important information about the files that make up the program. Instructions for creating the .INS file are found in the SDK [72].

The software can be distributed in several ways:

- on floppy disc
- on CD-ROM
- on the Memory Card
- on a Web site

As stated previously, the .INS file is needed when using any of these delivery methods. When distributing the application on a Web site, each software package must be bundled up into an Add-On Software file (.aos file) – a specially tagged archive. This file is used for the application installation to the communicator. The basic instruction for making an .aos file is:

1. Pack the package descriptor file, the application file (.geo file) and all the other files required into an aos-package. There is a specific Tar -program, which can be used for creating an .aos file. Tar is a standard Unix program and there is also a PC -version available. The host-file-location lines in the .ins file should describe locations within the tar file's directory structure. The .ins file must be in the top level directory of the tar file so that it may be found. Example of the tar command is:

   tar -cvf doc.aos doc.ins doc.geo

2. Give the tar file's filename the suffix .aos so that it will be recognised as an Add-On Software file.

3. Put the file up on your web server.

For more information about creating an .aos file, please see the SDK [73].

[72] pcgeos\Techdocs\Nokia9000\Distribution\index.htm
[73] pcgeos\Techdocs\Nokia9000\Distribution\Distribution_3.htm

Copyright © 1999 Nokia Mobile Phones.
All rights reserved.

22. SUMMARY

We hope that this document gives you the basic information necessary to develop a new application for the Nokia 9110 Communicator which will have, as much as possible, the same "look and feel" as the original applications produced by Nokia Mobile Phones for the Nokia communicators. By following these user interface design guidelines, it will make the new applications for the end-users easier to learn, easier to use and usage will be more error free. As a result the end-user will find the new application more productive and will be more satisfied with the application.

Nokia Mobile Phones has several services available to Nokia communicator software developers. One example of these is a web-based discussion forum for the developers. The Customer Application Development group grants Usernames and passwords. In some cases the forum is also used for document distribution. Nokia Mobile Phones, Customer Application Development group also regularly sends news for developers by e-mail. Nokia continues to develop its products and the information in this document is as accurate as possible at the time of writing. Nokia will update this document as needed.

Marketing help

Nokia is encouraging third-party software and application development for Nokia communicators. Please, inform us of your new applications so that Nokia can add them to the Application guide, where all the communicator applications are listed. The Customer Application Development group will be in charge of delivering this guide. Those applications, which have passed the Nokia certification test, will receive the Nokia OK logo. Nokia OK logo applications can be included on the CD-ROM supplied in new Nokia Communicator sales boxes. Also these applications can be promoted on Nokia Web sites. For more information, contact the Customer Application Development group [74].

Legal Notice

The content of this document is copyright © Nokia 1995, 1996, 1997, 1998, and 1999. All rights reserved. Reproduction, transfer, distribution or storage of part or all of the contents in any form without the prior written permission of Nokia is prohibited.

The NOKIA name and Nokia product names are trademarks of Nokia.

The content of this document is provided "as is". Except as required by applicable law no warranty of any kind, either express or implied, is made in relation to the accuracy, reliability or content of the document. Nokia reserves the right to revise the document or withdraw it at any time.

If you have any questions or comments about this document please send us feedback. Our e-mail address is:
9110.sw.developer@nokia.com

[74] 9110.sw.developer@nokia.com

Copyright © 1999 Nokia Mobile Phones. All rights reserved.