WAP Service Developer's Guide for Nokia 6210 and Nokia 6250

October 2000

Disclaimer:
Nokia Mobile Phones Limited disclaims all liability, including liability for infringement of any proprietary rights, relating to the implementation of information presented in this document. Nokia Mobile Phones Limited does not warrant or represent that such use will not infringe such rights.
Nokia Mobile Phones Limited retains the right to make changes to this specification at any time without notice.

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

Contents

1. INTRODUCTION 4
1.1 References 4
1.2 Contact information 4
2. WIRELESS APPLICATION PROTOCOL - WAP 6
2.1 Introduction 6
2.2 WAP Architecture 6
2.3 Developing Applications with WAP 7
2.4 Further Information 8
3. INTRODUCTION TO THE USER INTERFACE OF THE NOKIA 6210 9
3.1 Keypad - detailed description 9
3.2 Display - detailed description 9
3.3 WAP Browser Display 10
4. NOKIA 6210 USER INTERFACE ELEMENTS OF WML 14
4.1 Display and Formatting Elements 14
4.2 Images, links and timers 18
4.3 WML DO Construct 21
4.4 WML Input processing 23
4.5 Labels and Titles 25
4.6 Limitations 25
5. GENERAL USABILITY ISSUES – GUIDELINES FOR DESIGN 27
5.1 Mobile Applications For Mobile Use 27
5.2 Validate Your WML 27
5.3 Site Organization 27
5.4 Pictures, Tables and Colour 28
5.5 Take care of backward navigation 28
5.6 Use of card titles and element labels 29
5.7 Perform usability test 29

1. INTRODUCTION

This Service Developer's Guide for the Nokia 6210 and the Nokia 6250 provides information and practical examples for developers who want to develop WAP services for the Nokia 6210 and the Nokia 6250. The document will highlight the possibilities, limitations and requirements of service development and provides a good introduction to the Nokia style of designing user interfaces. Although the following chapters focus on the Nokia 6210 user interface, the guidelines can be applied to the Nokia 6250 as the browser is basically the same in both phones.

The Wireless Application Protocol (WAP) will enhance the functionality of mobile handsets through real-time interactive services. The protocol has been designed specially for small screens and low bandwidth, and it offers a wide variety of wireless services over the Internet using handsets. In addition to Nokia 6210 and Nokia 6250 -specific issues, many of the instructions in this document can be used to maximize interoperability and ease of use on various other browsers.

This guide is not written for service developers only, but essentially for anyone involved in creating the wireless information society and anyone who needs to know more about service creation on small mobile terminals. It can be used in conjunction with the Nokia WAP Toolkit.

The Toolkit offers developers an environment for creating, testing and demonstrating WAP applications. This allows service providers to evaluate the usability of wireless applications and services together with their end-user organization.

We at Nokia are very proud of our user interfaces and we have put a lot of effort into constantly improving them. In this guide we have collected a lot of what we have learned to help developers improve their services. We believe that good usability will increase use and satisfaction and subsequently revenues for service providers.

This guide is not intended to be a detailed WAP document and it is not meant to replace any WAP specification.

1.1 References

User's guide for the Nokia 6210

User's guide for the Nokia 6250

The Nokia WAP Toolkit can be downloaded from http://www.forum.nokia.com/

Wireless Application Specifications http://www.wapforum.org/

Information about Nokia products can be obtained from http://www.nokia.com/

1.2 Contact information

Developer support can be obtained through Forum Nokia web-page http://www.forum.nokia.com/

Terminology

Display
The word ‘display’ refers to the 4 lines of text (not including the Header text) that are visible at the same time in one view. You move from one display to another by rolling the roller down 4 times.

Card
A single WML unit of navigation and user interface.

Deck

A collection of WML cards that is the smallest download unit. You cannot download a single card in a deck; you must download the entire Deck. Service developers should be aware of this, and not do a giant application in one big Deck.

Hyperlink
A link within a document or card that allows quick navigation to another document (card).

Do element
A way of defining a binding between events (e.g. a user selection) and a task (e.g. to go somewhere).

Input element
A mark-up element that allows for interaction with the user, i.e. letting the user input textual values. These can be used for inputs to locally stored scripts, or for parsing to the server of origin.

Prev element
Lets the application developer define what happens, when the user selects Back. In some applications it might make more sense to jump to another card than the one that the user has just visited.

Options list
The Options list is the list of items shown when the user presses the Options soft key.

Proportional fonts
A font, in which the letters don’t necessarily take up the same amount of pixels in width. ‘W’ and ‘I’ are good examples.

Navigation
The principle of moving between menu items, decks and cards.

2. WIRELESS APPLICATION PROTOCOL - WAP

2.1 Introduction

The Wireless Application Protocol (WAP) is a set of protocols that allow the development of applications and services for use with Mobile Phones and other mobile devices. These protocols and their related standards and specifications are maintained by the WAP Forum.
The WAP Forum consists of a number of hi-tech companies from the Information Technology, Software and Telecommunications industries. The objectives of the WAP Forum are to:

- Bring Internet content and advanced data applications to digital cellular phones.
- Create a global wireless protocol specification that works across different wireless network technologies.
- Enable the creation of content and applications that scale across a wide range of bearer networks and device types.
- Embrace existing standards and technology wherever possible.

2.2 WAP Architecture

2.2.1 WAP and Internet Architecture

In order to leverage on the existing Internet standard as much as possible, the WAP stack closely follows the Internet model. This is illustrated in Figure 1.

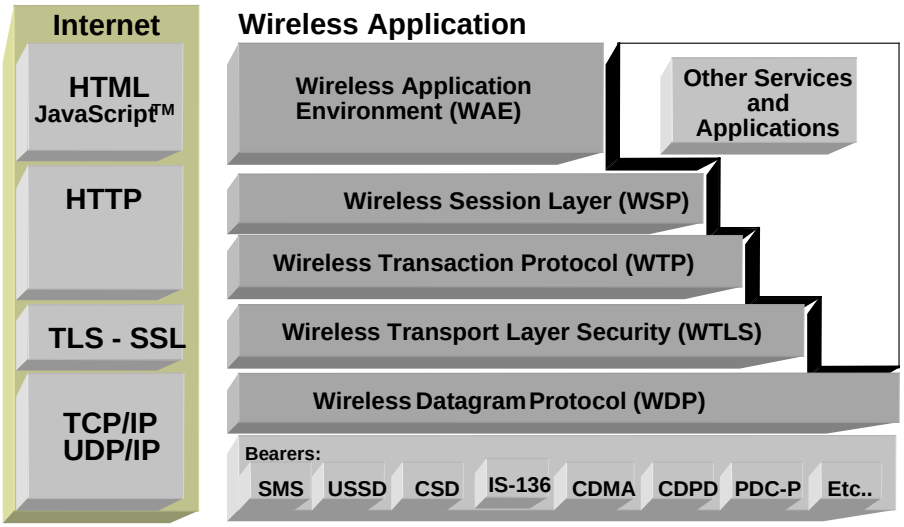

Figure 1 – Comparison of Internet and WAP Architecture

This layered architecture allows applications to utilize the features of the WAP stack through well-defined interfaces. This close link to the Internet architecture allows developers to utilize their existing knowledge and expertise when developing applications for mobile terminals.

2.2.2 Wireless Application Environment (WAE)

The Wireless Application Environment (WAE) is a general-purpose application environment that uses a combination of Internet and mobile terminal technology. It provides a framework for the development of applications on a mobile terminal.
The WAE contains support for the following functionality:

Wireless Mark-up Language (WML) - a lightweight presentation language, similar to HyperText Mark-up Language (HTML) but optimized for use with mobile terminals.
Wireless Mark-up Language Script (WMLS) - a lightweight script language, similar to Java Script™.
Wireless Telephony Application/Interface (WTA/WTAI) - telephony services and programming interfaces.
Content Formats - defined data formats, such as vCard and vCalendar.

2.2.3 Wireless Session Protocol (WSP)

The Wireless Session Protocol (WSP) provides the application layer of the WAP with an interface for two session services. The first is a connection-oriented service that operates above the transaction layer protocol. The second is a connectionless service that operates above a secure or non-secure datagram service.
The WSP is optimized for low bandwidth bearer networks with long latency.

2.2.4 Wireless Transport Protocol (WTP)

The Wireless Transport Protocol (WTP) runs on top of the datagram service and provides a lightweight transaction-oriented protocol suitable for use in mobile terminals. WTP operates over secure or non-secure wireless datagram networks.

2.2.5 Wireless Transport Layer Security (WTLS)

The Wireless Transport Layer Security (WTLS) is based on the industry standard Transport Layer Security (TLS) and is optimized for use over narrow band communication channels. WTLS may be used for secure communication between terminals, and applications can selectively enable WTLS features.

2.3 Developing Applications with WAP

Application Developers can use the principles of WAP to develop new services or adapt existing Internet applications for use with mobile terminals. Applications are written in the Wireless Mark-up Language (WML) and the WMLScript, and stored on either a normal web server (origin server) or directly on the WAP Gateway. The content stored on the web server will be accessible from the mobile terminals via the cellular network and a WAP gateway or proxy.
The Proxy Server acts as a gateway between the cellular network and the Inter/Intranet. The data sent between the origin server and the handset is binary encoded to optimize transmission over the narrow bandwidth of the cellular network. Note that the content stored on the web server might be in either textual or binary format. When the WAP gateway fetches textual content, it automatically compiles this to the encoded format to minimize network load.
Figure 2 shows the network-related elements required for developing and offering services to mobile users.

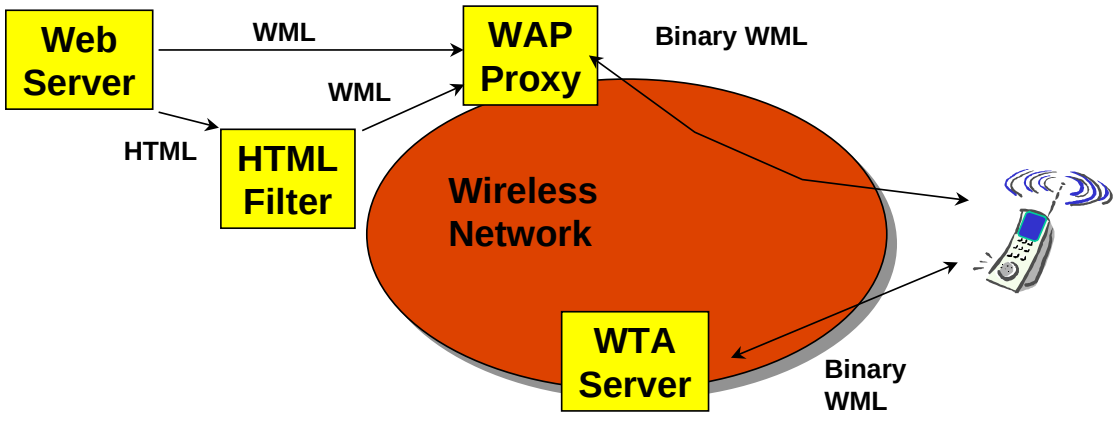

Figure 2 – Network-Related Elements for WAP Applications

Web Server:
The Web Server stores the applications written in WML. Alternatively, existing HTML applications can be used.
HTML Filter:
Any applications written in HTML will have to be converted to WML before they are sent to the mobile terminal. This HTML Filter may form part of the Web Server or the WAP Proxy.
WAP Proxy:
The WAP Proxy acts as the gateway between the cellular network and the Inter/Intranet. It binary encodes the information which is to be sent to the mobile terminal and decodes information sent from the mobile terminal.
WTA Server:
The WTA server handles network-specific applications. These applications are not discussed in this document.

2.4 Further Information

Further information about WAP is available on the Internet at www.wapforum.org.

3. INTRODUCTION TO THE USER INTERFACE OF THE NOKIA 6210

This section gives a short overview of the user interface of the Nokia 6210 phone. The Nokia 6210 phone is a mobile phone supporting many features, like Calendar, Advanced Messaging, Phone Book, etc. The WAP Services (Browser) part of the phone is designed to comply with the user interaction handling known from other applications in the phone. This section provides a basic understanding of how the product behaves when using different kinds of Browser/WML elements.

3.1 Keypad - detailed description

Keypad Design and layout of Nokia 6210

Numeric Keys, The numeric keys basically have no function when viewing cards or in the Options lists. The Numeric keys are used in the editor, as follows:

When a number input has been requested via the WML, the numeric keys will insert the corresponding digit (and */#)

The **# key** is used for entering SS strings and for changing character case during editing.

The *** key** is used for entering SS strings, and for entering international numbers.

The **Send/talk key** works as a select key.

The **End key** exits a data call and goes to the idle state when pressed twice.

Two Soft keys. The soft-keys are assigned actions, that enable the user to manipulate the user interface by making selections, and entering, editing and deleting text.

- The **Left soft key** is basically used as a yes/positive key. It will contain options that execute commands and go deeper into the menu structure: Select, OK, Options and similar.
- The **Right soft key** is basically used as a no/negative key. It will contain options that cancel commands, delete text and go backwards in the menu structure, Back, Exit, Clear.

Scroll keys allow the user to scroll through the options or text in the current display.

3.2 Display - detailed description

The Nokia 6210 display is a full dot matrix display with a display resolution of 96 pixels (horizontal) by 60 pixels (vertical). The display consists of the application area with header text and the area used for the soft keys. See Figure 3-1 for a description of different parts of the display.

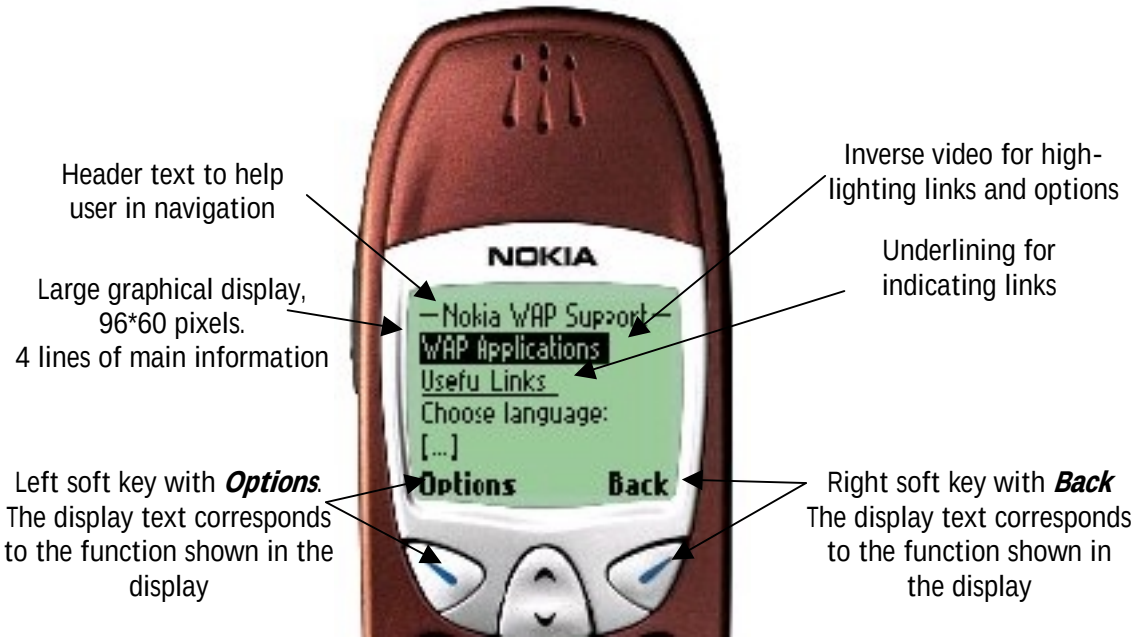

Figure 3-1 Different parts of the display

3.2.1 Physical dimensions of the browser screen

The application area is 96 (horizontal) by 52 pixels (vertical); it includes 4 lines of main text, plus a header text. The main text area can also be used for showing graphics. The graphics area is at maximum of 96 pixels wide and 41 pixels high but scrollable until maximum deck size (i.e. an image can be scrolled down like any other content).

3.2.2 Display Fonts

Font size in the Nokia 6210 is 8 pixels plus 2 pixels line break. All fonts used in the Nokia 6210 are proportional fonts. Proportional fonts give a dynamic and minimized width for each character, they give improved readability of the display texts, and they generally allow more characters to be displayed per line. However, having proportional fonts means that it is difficult to tell how many characters there can be in each line cf. Characters 'W' and 'i'.

3.2.3 Word Wrapping

Word wrapping is also important when designing services. If the first two words of text do not both fit into the first line, the second word will automatically be moved to the second line. However in the Nokia 6210 the word wrapping can be set of in the browsers appearance settings. Using the Toolkit is one method of checking how the text will be shown on the display.

3.2.4 The Header Text in the Nokia 6210

The Header text will contain the `title` attribute in the WML card. This header text can help the user to know where they are in the service, and allow ease of navigation in the browser.

3.3 WAP Browser Display

The WAP Browser main view opens when selecting the 'Services'-menu

The Home button fetches information from the homepage. The homepage is related to the used access point. The access point and homepage can be set in Settings.

3.3.1 Scrolling

Scrolling selectable elements

Whenever there are selectable elements in the display, the roller key will allow the user to scroll between the selectable items. Scrolling down in the above example will give

Figure 3-2

Figure 3-3

Selecting a highlighted element is done by pressing the send/call-key.

Scrolling plain text

If the user scrolls through plain text in the browser display, the text is scrolled line by line. Scrolling down gives one extra line of the text for example:

Figure 3-4

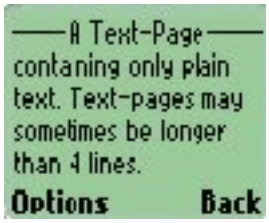

Figure 3-5

3.3.2 Options on the left soft key

The left soft key will, in all normal browsing situations, contain 'Options' which leads to the 'Service options' list. The service options list is a mixed list, containing both context-sensitive options (like all the Do-elements), and also ‘fixed’ browser options (Bookmark handling, etc). The contents and ordering of the options are listed below and are followed by a more detailed description of the fixed browser options. Context-sensitive options are included in chapter 5. Nokia 6210 user interface elements of WML.

- *Home* (short-cut to the Home page)
- *Bookmarks* (see section 4.9 Bookmark handling).
- *Select* or *Edit* (works as go -element for links and Select for selection lists or links), works as ‘Edit’ for input -element.
- do Element number 1 (provided by the WML Card - if available)
- do Element number 2 (provided by the WML Card - if available)
- do Element number X (provided by the WML Card - if available)
- *Use Number* (will allow the user to call (or edit) any of the numbers in the current Card). Only available if there are numbers in the Card. See also section 3.3.4 Use Number handling).

- *Empty Cache* (Removes all cached WML Decks in the phone).
- *Exit* (Exits the browser, ending up in the main menu with the Browser item highlighted. With confirmation)

3.3.3 Bookmark handling

The Bookmark handling of the Nokia 6210 is entered using the *Bookmarks* item contained under the options in the left soft key, *Options*. The user can save URLs as bookmarks while browsing on that page, by adding them manually or by saving a bookmark that has been sent as an over the air message. The maximum amount of bookmarks that can be stored is 15.

When entering the *Bookmark*s option, the user will get access to the currently defined bookmarks. This could look like:

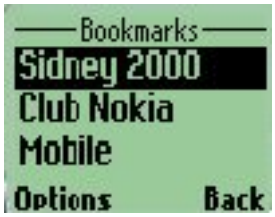

Figure 3-6

The user can select a Bookmark by pressing the left soft key. The Options in the left soft key in the Bookmarks list will contain:

- *Save bookmark.* Allows the user to add the current Card to the bookmarks list.
- *Add Bookmark*. Type in and add a bookmark to the bookmarks list.

When there is at least one bookmark in the bookmarks list, the following will also be shown:

- *Go to*. An alternative way of selecting the highlighted bookmark.
- *Edit*. Allows the user to edit the highlighted bookmark, changing the URL and title.
- *Erase*. Allows the user to erase the highlighted bookmark.
- *Send bookmark*. Allows the user to send bookmarks via SMS to other WAP phones that support the WAP OTA specification.

3.3.4 Use Number handling

The *Use Number* option is available via the ***Options*** soft key in the normal Browser display (when viewing a specific card). A "No numbers found" message is displayed if the item is used and no numbers exist on the active card. The *Use Number* option allows the user to select any of the numbers in the card

Figure 3-7

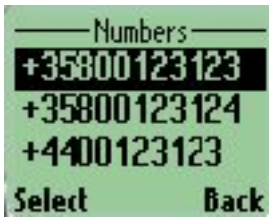

Figure 3-8

The user can scroll through the list using the scrollkey and select a number by pressing the *Select*-soft key. Selecting a number will exit the browser and paste the number into the normal phone Standby Mode. From the Phone Standby Mode, the user can call the number, save it or send a short message directly to it.

4. NOKIA 6210 USER INTERFACE ELEMENTS OF WML

This chapter is a guide to using WML in designing services specifically for the Nokia 6210. It is an overview of the graphical user interface elements, i.e. the WML elements that are critical for the rendering point of view and their WML capabilities supported by the browser. The document doesn't include all possible WML elements and attributes. The WML Version supported is v.1.1

The elements are briefly described and shown in example figures and WML code. The code examples include only the essential parts required in the using of the elements, i.e. the document prologue is not shown after example 1.

4.1 Display and Formatting Elements

The application area in the Nokia 6210 (described in more detail in chapter 3.2) is 96 (horizontal) by 52 pixels (vertical); it includes 4 lines of main text, plus a header text. The main text area can also be used for showing graphics. The graphics area is a maximum of 41 pixels high and 96 pixels wide.

Figure 4-1 WML Browser's Card View

4.1.1 Card

The contents of the WML card are displayed in the card view (See Figure 4-1 above). The card element is a container of text and input elements and indicates the general layout in the WAP browser. The order of elements in the card is significant, as they appear on the screen in the respective order of the card.

4.1.2 Paragraph and text alignment

Paragraph <p> determines the parts of the text in a card; a new paragraph always starts on a new line. Sentences thet are too long to fit across the screen are in all cases word-wrapped.

Text inside a paragraph can be aligned left, center or right by option; left is displayed by default. Alignment is determined in paragraph attributes (f.ex. <p align="right">). Figure 4-2 below illustrates how text can be located in a card view.

Figure 4-2 Text and image aligning

Example 4-1 Align <align>

```
<wml>
<card id="card1" title="Align" ordered="true"
newcontext="false">
<p align="center">
Align &quot;center&quot;
</p>
<p align="right">
Align &quot;right&quot;
</p>
<p align="right">
<img alt="8210" src="file:///C|/USERS/wml/8210.wbmp"
align="top" hspace="0" vspace="0"/>
</p>
<do type="previous" label="Back" optional="false">
<prev/>
</do>
</card>
</wml>
```

4.1.3 Line Break

New lines in the text can be defined by a line break. The line break
 -element can be used inside other elements too, as long as it belongs to a paragraph.

Figure 4-3 Line break in text

Example 4-2 Line break

```
<wml>
<card id="card1" title="Break" ordered="true"
newcontext="false">
<p align="left">
Here's a page containing only plain text.
<br/>
Text-pages may sometimes be longer than 4 lines
</p>
<do type="previous" label="Back" optional="false">
<prev/>
</do>
</card>
</wml>
```

4.1.4 Fieldset

Fieldset determines the text groups; fieldset always starts a new line and ends with a line break.

4.1.5 Tables

The table element is used to create columns and rows of text and images in a card surrounded by a grey border. The WML element of the table is `<table>`, and its content is specified by `<tr>` and `<td>`. Table content may contain text, images or links. Table elements do not specify column or intercolumn widths; the width of the column is the same as the width of the widest cell in the column. If the cell content is too long to be displayed in one row, the text is automatically word wrapped to the required amount of rows to fit the cell. A line break is inserted before and after the table. The table may contain links; they are focused one row at a time from top to bottom when scrolling down with arrow keys (Figure 4-5). In the first phase a focused row is selected and in the second phase the specific link is selected (Figure 4-6).

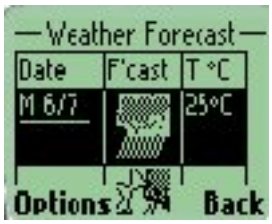

Figure 4-4

Figure 4-5

Figure 4-6

Example 4-3 <table>

```
<wml>
   <template>
   <do type="previous" label="Back">
     <prev/>
   </do>
   </template>
<card id="card1" title="Weather Forecast">
<p align="left">
<table align="LCC" columns="3">
  <tr>
    <td>Date</td>
    <td>F'cast</td>
    <td>T °C</td>
  </tr>
  <tr>
    <td>
    <anchor title="date">M 6/7
    <go href="#date" method="get" sendreferer="false"/>
    </anchor>
    </td>
    <td>
    <anchor>
    <img alt="rain"src="file://…/rainy.wbmp"
    align="bottom"/>
    <go href="#descr" method="get" sendreferer="false"/>
    </anchor>
    </td>
    <td>25°C</td>
  </tr>
  <tr>
    <td>T 6/8</td>
    <td>
    <img alt="part cldy" src="file://…/partcldy.wbmp"
    align="bottom"/>
    </td>
    <td>27°C</td>
  </tr>
  <tr>
    <td>W 6/9</td>
    <td>
    <img alt="cloudy" src="file://…/cloudy.wbmp"
    align="bottom"/>
    </td>
```

```
    <td>24°C</td>
  </tr>
  <tr>
    <td>T 6/10</td>
    <td>
    <img alt="rainy" src="file://…/rainy.wbmp"
    align="bottom"/>
    </td>
    <td>28°C</td>
  </tr>
  <tr>
    <td>F 6/11</td>
    <td>
    <img alt="sunny" src="sunny.wbmp" align="bottom"/>
    </td>
    <td>29°C</td>
  </tr>
</table>
</p>
</card>
```

```
<card id="date" title="Date" ordered="true"
newcontext="false">
<p align="left">Week 23/2000
</p>
</card>
```

```
<card id="descr" title="Description" ordered="true"
newcontext="false">
<p align="left">
Cloudburst, deluge- and storm warning
</p>
</card>
```

```
</wml>
```

4.2 Images, links and timers

4.2.1 Img – Images

The image element <img> is used for showing an image in a card. The Nokia 6210 supports image format Wireless Bitmap (wbmp). The maximum size of the graphics area is 96x44 pixels. The alt-text (used as an attribute) is shown on the screen during the load time or if the image cannot be displayed at all. However, in the Nokia 6210, showing images is optional and can be turned off in the browser's appearance settings. Image sizes can be 1.3 Kb or less; there are no actual limits to the physical measurements of an image; however, when the user scrolls down bigger images only the leftmost part is shown for images wider than the browser area. Images can also be used inside a table and as a link (see Example 4-3 above).

Image sizes of a maximum of one line, i.e.11 pixels high may be shown with three lines of text on the same display above and/or below the image. Similarly, with an image of a maximum of 22 pixels there can be two lines of text (see the example below), and one line of text with an image of a maximum of 33 pixels. There cannot be text next to an image; the image will always start its own line as shown in the following example:

Figure 4-7 Image and text

Images fitting into the graphics area are centred. Images that are larger than the display are left-aligned and truncated from the right-hand part (see Figure 4-8). Images that are taller than the display are top-aligned, and the bottom part of the display is truncated but can be scrolled down (see Figure 4-9).

Figure 4-8 Images too wide are truncated

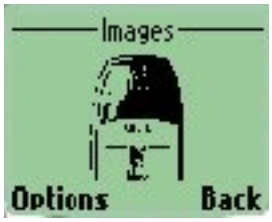

Figure 4-9 Images too high can be scrolled

If a requested image does not exist, a small default picture will be shown. We recommend using the image ALT attribute value.

Example 4-4 <img>

```
<wml>
<card id="card1" title="Images">
<p align="center">
A small image
<img alt="part cldy" src="file://…/partcldy.wbmp"/>
displayed with text
</p>
<p align="center">
<img alt="8210"
src="file:///C|/USERS/6210doku/WML/8210.wbmp"/>
<img alt="nokia"
src="file:///C|/USERS/6210doku/WML/120nokia3.wbmp"/>
</p>
<do type="previous" label="Back" optional="false">
<prev/>
</do>
</card>
</wml>
```

4.2.2 Anchors

Use the <anchor> element to create a link. A link provides easy navigation through an application and lets the user navigate to a new location. A link can be either text or an image. The <a> element is

a short form of the <anchor> element, and is bound to a go task without variables. In general it is advisable to use the <a> element instead of <anchor> where possible, to allow more efficient tokenisation.

In Example 4-5 there is a text link and an image link to another card in the deck. The user activates a link by selecting the link. Anchored links are as underlined, and when selected are seen with a highlight emphasis.

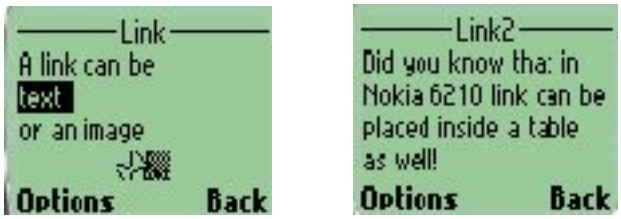

Figure 4-10 Links can be text or image

Example 4-5

```
<wml>
```

```
<template>
  <do type="previous" label="Back" optional="false">
  <prev/>
  </do>
</template>
```

```
<card id="card1" title="Link">
<p align="left">
  A link can be
<anchor>text
<go href="#card2"/>
</anchor>or an image
</p>
```

```
<p align="center">
<anchor>
<img alt="part cldy" src="file://…/partcldy.wbmp"/>
<go href="#card2"/>
</anchor>
</p>
</card>
```

```
<card id="card2" title="Link2">
<p align="left">
Did you know that in the Nokia 6210 links can be placed
inside a table as well!
</p>
</card>
```

```
</wml>
```

4.2.3 Timer

The timer element declares a card timer, which exposes a means of processing inactivity or idle time. This element can be used only once in a card and its unit is 1/10s.

4.3 WML DO Construct

The <do> -element binds a task to a user action, i.e. it is used for adding an action or general information links which concern the whole service. When you add a do-element to your design, it will be stored in the browser's options list together with the preloaded options like home, bookmarks, use number etc. A prev-type do-element, i.e. Back-button usage is highly recommended because it enables backward navigation. The prev is displayed in the right soft key as well as in the Service options. Prev can be implemented in two different ways: card wide or deck wide. In card wide prev the element is placed inside a card; in deck wide prev there is a template determining the prev-functionality for every card in a deck.

4.3.1 Prev element

Figure 4-11

The author of the WAP service is always responsible for the working navigation model inside its own service. If there is a need for the back functionality it should be explicitly defined with the PREV type and DO element. The WAP browser does not have any implicit back functionality!

Example 4-6 Back navigation <prev>

```
<template>
<do type="previous" label="Back" optional="false">
<prev/>
</do>
</template>
<card id="card1" ordered="true" newcontext="false">
<p align="left">
Always include a prev-element in a card to enable backward
navigation!
</p>
<do type="previous" label="Back" optional="false">
<prev/>
</do>
</card>

</wml>
```

4.3.2 Do element

The do element can be expressed in two ways: in card wide scope (like 'Enter' in Example 3-7) or in deckwide scope as in the template (like 'Ask Help', 'Contact' and 'Back' in Example 3-7). Do elements will be stored in the browser's options list together with the preloaded options (Figure 4-13).

Figure 4-12

Figure 4-13

Example 4-7 do-element

```
<wml>
<template>
<do name="help-button" type="options" label="Ask Help">
<go href="http://.../help.wml" method="get"/>
</do>
<do type="options" label="Contact">
<go href="http://.../contact.wml" method="get"
sendreferer="false"/>
</do>
<do type="prev" optional="false">
<prev/>
</do>
</template>
<card id="card1" title="Homepage" ordered="true">
<p align="left">
Please, register
<br/>
Username:
<input name="username" type="text" title="Username"
emptyok="false"/>
Password:
<input name="password" type="password" title="Password"
emptyok="false"/>
<do type="accept" label="Enter">
<go href="#confirm" method="get" sendreferer="false"/>
</do>
</p>
</card>
<card id="confirm" title="Confirmation" ordered="true"
newcontext="false">
<p align="left">
$(username), thank you for registering.
</p>
</card>
</wml>
```

4.4 WML Input processing

Input processing enables the user to input requested information into the service. There are two kinds of input elements: text fields determined by input-elements and selection lists determined by select-elements. Option elements are used to specify a single choice option in a select element. Option elements can be grouped by using the optgroup-element (see Example 4-7).

4.4.1 Input elements

Figure 4-14

Figure 4-15

Figure 4-16

The input element specifies a text entry object. The text editor can be opened by pressing the Send/Talk-button. Input fields can be specified by input-elements type and format attributes.

Example 4-8 Text input <input>

```
<template>
<do type="previous" label="Back" optional="false">
<prev/>
</do>
</template>
<card id="card1" title="Homepage">
<p align="left">
Please, register
<br/>
Username:
<input name="username" type="text" title="Username"/>
Password:
<input name="password" type="password" title="Password"/>
<do type="accept" label="Enter" optional="false">
<go href="#confirm" method="get" sendreferer="false"/>
</do>
</p>
</card>
```

```
<card id="confirm" title="Confirmation" ordered="true">
<p align="left">
$(username), thank you for registering.
</p>
</card>
```

```
</wml>
```

4.4.2 Select elements

Selection list (<select><option>) is an input element that specifies a list of options from which the user can choose. There are two kinds of selection lists: single choice and multiple choice lists. The user can select the multiple-choice options, if the multiple attribute is set as TRUE (selected items are marked with a selection symbol).

The figure below (4-15) shows how the multiple selection element is rendered. When the input item is selected the list of options appears. List items are marked/unmarked by pressing 'mark/unmark'-soft button. When the 'done' –soft button is pressed the selections are moved to the input box on the deck. The user can see the selections on the box. If those strings cannot fit the box, only the first ones are shown.

Options can be grouped with the <optgroup> element. The group title is essential as it represents the subgroup items. After selecting a list by pressing the 'list'-soft button, the functionality is similar to selecting multiple items from the selection list. In Example 4-8 there are two option groups. Item is selected first from the "Phones"-list (see Figure 4-15) and then from the other list of options; "Accessories" (see Figure 4-16).

Figure 4-17 Selecting from the group 'Phones' using an optgroup list.

Figure 4-18 Selecting from the group 'Accessories' using an optgroup list.

Example 4-9 Select and optgroup

```
<wml>
<card id="phones" title="Information Request">
<p align="left">
Send me more information about:
<select name="models" title="Products" multiple="true">
  <optgroup title="Phones">
  <option value="Nokia 8210">
    8210
  </option>
  <option value="Nokia 7110">
    7110
  </option>
  <option value="Nokia 6110">
    6210
```

```
  </option>
  </optgroup>
  <optgroup title="Accessories">
  <option value="Battery">
    Battery
  </option>
  <option value="Desktop stand">
    Desktop stand
  </option>
  <option value="Charger">
    Charger
  </option>
  <option value="Headset">
    Headset
  </option>
  </optgroup>
</select>
Please, provide  your
<br/>
Name:
<input name="name" type="text" title="Name"
emptyok="false"/>
</p>
<do type="accept" label="Confirm" optional="false">
<go href="#confirm" method="get" sendreferer="false"/>
</do>
<do type="prev" optional="false">
<prev/>
</do>
</card>
</wml>
```

4.5 Labels and Titles

Please provide the user with the following titles and labels:

The `title` of a card to help him/her to navigate in your service.

The `title` of a `select` element is used in the header of the selection list or option groups list.

The `title` of an `optgroup` element is used as an option group's title and in the header of its selection list.

The `title` of an `input` element is used in the editor title.

The `label` of a <do> element is used in the Service options behind the 'options'-soft button.

4.6 Limitations

This section includes the technical limitations of the WAP services application.

The total cache size in the Nokia 6210 is 50 Kb.

The maximum item size is about 1.3 kilobytes binary. This is the maximum size both for an encoded WML deck and for an image. The WML deck and images are downloaded separately.

4.6.1 Rendering Limitations

Emphasis tags are not supported

Links are always on their own line

Images are always on their own line

Input fields are always on their own line

Vertical alignment is not supported

5. GENERAL USABILITY ISSUES – GUIDELINES FOR DESIGN

Nokia phones are renowned for their ease of use and intuitive user interface. In order to create a service that will be perceived as usable and easy to understand, Nokia Mobile Phones provides some general guidelines to help service providers develop their applications.

5.1 Mobile Applications For Mobile Use

When deciding what information to include in the different applications on a mobile terminal, think about which information might be relevant in the situations where the Mobile Phone will be used. Because of the mobility of the small display, the user might primarily use the Mobile Phone when there is no access to the Internet on a PC. We believe that users of the mobile applications will primarily be interested in brief and quick information. For instance, quick access to flight schedules from the Mobile Phone might be relevant for a user. Similarly, several short news flashes are more relevant than longer news articles. Quick access to weather information may also be of use in the mobile situation to quickly check out the temperatures of the destination of the traveller. However, it is less likely that users will surf on the application on the mobile terminal, but will prefer to surf the Internet on a PC from their office or home.

5.2 Validate Your WML

There are several XML validators available that validate your documents against WML Document Type Definition. It is recommended that authors validate their WAP pages, because invalid WML is always treated as an error and discarded (ie. not shown to the user!).

Several XML validators can be found at http://www.wapdevelopers.org/xml.tpl?CALLER=index.tpl.

5.3 Site Organization

5.3.1 Avoid ‘doormat’ pages

The user is accessing your WAP site over a GSM data call, and pays per second. It is not recommended that you start your site using a ‘doormat’ page, which has no functionality other than perhaps to greet the visitor and to display a logo. It is better to go to your service directly. If there's a need to use the timer it shouldn't be timed for longer than 1.5 s, otherwise add a 'continue' link to give the user the possibility of exiting the card.

5.3.2 Automatically Detect the Browser

If you are using a modern web site hosting environment, it is possible to detect the browser as well as the language mode and supply the correct content transparently, without user interaction.

5.3.3 Optimize for Size

The size of the content is critical. If you have large decks (listings, large tables, etc.), consider splitting them into multiple parts for faster downloading.

As for the total download time, some studies place an upper limit for acceptable delay at 10 to 15 seconds, including all images, on a PC-based browser. It is highly recommended that there be a response time (excluding connection time) of less than 10s in a mobile environment.

5.3.4 Choose Descriptive Card Titles

It is very useful to give a descriptive name to the card. It might be a good idea to start the title with your service's name and keep the total length of the title short.

It also pays to use meaningful URLs since the user sees the URL of the currently selected link on the screen and can use it as a navigational aid, especially when images have not been loaded.

5.3.5 Pay Attention to the First Screen

The first (topmost) screen of any page is the most important one. All of the often-used navigational links, search fields, login screens, and the bulk of the information should reside there if at all possible. The user is then able to navigate forward before the rest of the card has been loaded, and the user does not have to scroll through the card.

Avoid wasting the top of the page on banner advertisements or non-informative graphics. It is better to place the advertisements at the left or right edges than on the top.

5.3.6 Do not Use Absolute Values for the Screen Size

When using images the use of absolute values (in pixels) is not recommended. Sizes should be specified as percentages of their total width or height.

5.3.7 DO-elements Usage

Always use descriptive and short labels for all DO-elements. Favour local DO-elements instead of anchored links that are out of context. Global DO-elements should be used only when especially needed at the end of the card. Always include the prev element in every card to enable backward navigation.

5.4 Pictures, Tables and Colour

5.4.1 Avoid Useless Images

Downloading of images takes time, and many users may switch the loading of images off for more speed. Try to optimize the size of the images. If you have large pictures on your site, consider using thumbnails for the image index.

Always give an alternative text (using the ALT attribute of the IMG element) for images that convey information. Always use a null alternative text (ALT="") for images which do not convey information, or are used for page layout or decorative purposes only.

5.4.2 Use Reasonable Tables Sizes

If the table size exceeds the maximum width of the application screen due to the number of columns the table size will be scaled down to fit the screen. To keep the cell content readable, special attention should be paid to the table structuring.

5.5 Take care of backward navigation

Due to the differences in the processing of the History between WAP and WEB, only the history processing described in the WML specifications is followed. This means that the service provider has to take care of the navigation as a whole!

5.6 Use of card titles and element labels

Card titles describe the content of the display and their use is recommended. They help the user to navigate in the application because they function as a reminder of where the user is in the application. The header text should be determined by the item previously selected by the user. For instance, the card title 'Bookmarks' tells the user that the display contains a list of bookmarks in the application and that the Options item previously selected was *Bookmarks*.

Proportional fonts are used in the header texts, and if the header text is too long it is automatically truncated. Truncation is usually better than abbreviations, because the user might be confused by unfamiliar abbreviations that can be difficult to understand.

5.7 Perform usability test

It is always good to perform a usability test on new applications. People who have not been involved in the design or development of the applications tend to notice potential usability problems often not obvious to those who know the design by heart. Usability tests should always be performed as early as possible in the development process. Any necessary changes resulting from the tests can then be implemented within the development timescale. Try to recruit users who are representative of the end users of the application, and try to conduct the usability test on a smaller scale, if the timescale does not allow for extensive testing.