

WAP SERVICE DEVELOPER'S GUIDE FOR NOKIA SERIES 40

Version 1.0
13-02-02

Table of contents

1	INTRODUCTION	4
1.1	References	4
1.2	Contact information	5
2	GENERAL USABILITY ISSUES – GUIDELINES FOR DESIGN	6
2.1	Mobile applications for mobile use	6
2.2	Validate your WML.....	6
2.3	GPRS usability	6
2.4	Site organization	7
2.5	Pictures and tables.....	8
2.6	Take care of backward navigation	9
2.7	Use of card titles and element labels.....	9
2.8	Performing a usability test.....	9
3	INTRODUCTION TO THE USER INTERFACE.....	10
3.1	Nokia Series 40 phones.....	10
3.2	User interface – display and keys.....	11
3.3	Display fonts	12
3.4	Accessing WAP browser	12
3.5	Navigation, labels and titles.....	12
3.6	Scrolling	12
3.7	Picture viewer	13
3.8	Cache	13
4	WML ELEMENTS FOR THE NOKIA USER INTERFACE	14
4.1	Text formatting	14
4.2	Table	15
4.3	Images	17
4.4	Links	18
4.5	Do element	18
4.6	WML input processing.....	20
5	OPTIONAL WAP 1.2.1 FEATURES	24
5.1	Push Service Indication.....	24
5.2	WTAI public library – Wireless Telephony Applications Interface public library	24
5.3	Accesskey	25
5.4	Cookies	26
5.5	Content download over WAP	26
A.	WIRELESS APPLICATION PROTOCOL WAP.....	27
	Introduction	27
	WAP architecture.....	27
	Developing applications with WAP.....	28
	Further information.....	29
B.	LIST OF TERMS AND ABBREVIATIONS.....	30

WAP Service Developer's Guide for Nokia Series 40
Version 1.0

Change history

13-03-02	Version 1.0	Document added into Forum Nokia
----------	-------------	---------------------------------

Disclaimer:

Nokia Corporation disclaims all liability, including liability for infringement of any proprietary rights, relating to the implementation of information presented in this document. Nokia Corporation does not warrant or represent that such use will not infringe such rights.

Nokia Corporation retains the right to make changes to this specification at any time without notice.

Licence:

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

1 INTRODUCTION

This WAP Service Developer's Guide provides information and practical examples for developers who want to optimise their existing WAP services for the Series 40 phones. Series 40 is a UI category for Nokia 128 x128 pixel screen resolution. It comprises a passive matrix colour screen with two soft keys. The current Nokia handsets that belong to this category are introduced in Chapter 3. This document highlights the possibilities, limitations, and requirements of service development, and provides a good introduction to the Nokia style of designing user interfaces.

The Wireless Application Protocol (WAP) enhances the functionality of mobile handsets through real-time interactive services. The protocol has been designed especially for small screens and low bandwidths, and it offers a wide variety of wireless services over the Internet for handsets. It was also designed to allow content to be delivered over any bearer service, even when delivery of the services is enabled over GPRS, 3G or any other type of network. WAP over GPRS opens up new possibilities for application development, and there are also some optimizations in GPRS, that can be done by service developers.

Chapter 2 describes some general usability issues concerning WAP service development. Chapter 3 introduces the browser user interfaces elements, keys, display and browsing conventions. Chapter 4 outlines some general WML elements, which are mandatory features on WAP 1.2.1, according to WAP Forum specifications. Chapter 5 introduces some optional WAP 1.2.1 features that are supported in most of the WAP1.2.1 compliant phones.

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

The Nokia Mobile Internet 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. 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 thus increase revenues for service providers.

This guide does not purport to describe WAP in detail, and it is not meant to replace any WAP specification.

1.1 References

User's guide for Nokia 7210

WAP Characteristics list can be found at <http://www.forum.nokia.com/>

The Nokia Mobile Internet Toolkit can be downloaded from <http://www.forum.nokia.com/>

GPRS related documentation could be found at <http://www.forum.nokia.com/>

WAP Service Developer's Guide for Nokia Series 40
Version 1.0

Wireless application protocol specifications can be found at
<http://www.wapforum.org/>

Information about Nokia products can be found at <http://www.nokia.com/>

1.2 Contact information

Developer support for WAP service development and GPRS bearer technology can be obtained through the Forum Nokia Web site at <http://www.forum.nokia.com/>

2 GENERAL USABILITY ISSUES – GUIDELINES FOR DESIGN

Nokia phones are renowned for their simple-to-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.

2.1 Mobile applications for mobile use

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

2.2 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 is discarded by the WAP gateway (that is, it is not shown to the user).

Several XML validators can be found at <http://www.wapdevelopers.org>

2.3 GPRS usability

GPRS features efficient use of resources, instant access, fast delivery of information and innovative charging models. Together, WAP and GPRS not only improve the user experience of existing applications but also enable new services.

2.3.1 Not necessary for developers to make changes

GPRS enhances the end user experience of the mobile services currently being delivered over circuit switched data (CSD). Since the WAP standard was designed to allow content to be delivered over any bearer service, it is not necessary for developers to make changes to their WAP-compliant services or applications, even when delivery of the services is enabled over GPRS, 3G or any other type of network.

2.3.2 Cache utilization

It is recommended to use HTTP proxy to cache WML content whenever the content is accessed through the Internet. The proxy should be placed close to the WAP gateway to minimise the distance for data transfer between the two components. The home page (first page) might alternatively be kept on a local server, close to the WAP gateway.

Most clients have a cache and using it is an important way to provide fast user experience.

2.3.3 Usability tips for WAP over GPRS service development

Round-trips in a network always cause a minimum of a two 2-second delay to the response time. Therefore, the number of round-trips should be minimised. For example, bitmaps cause additional round-trips. The response time can also be minimised by keeping the size of a WAP deck as compact as possible; this is especially important for pages that require an instant response. The response time of the first deck has a strong impact to on the user experience. Therefore, it is worthwhile to maximise Infrastructure optimisation.

Minimise latency between the WAP gateway and content server. Remember that each query traverses this link twice, once as a request and once as a response.

Place the WAP gateway close to the GPRS gateway. (The GPRS gateway is technically called a "Gateway GPRS Support Node", or GGSN.) "Close" in this instance refers to both latency and probability of packet loss. Lost messages cause additional delay due to HTTP-retransmission.

2.4 Site organization

2.4.1 Avoid "doormat" pages

The user accesses your WAP site over a GSM data call, and pays per second. It is not recommended to start your site using a "doormat" page, which serves no purpose other than perhaps to greet the visitor and to display a logo. It is better if the user can go to your service directly. If there is a need to use a timer, it should not be longer than ~1.5 seconds. Otherwise, add a link incorporating the text "continue" to give the user the possibility to exit the card.

2.4.2 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 to supply correct content transparently without user interaction.

Nokia WAP phones send a user agent field in the WSP header to the WAP Gateway when they fetch content from the origin server. You can use this user agent field to define the browser type and then build the logic on the origin server to serve suitable content. From the WAP Gateway, it is then delivered inside the HTTP header.

The user agents for different Nokia phones can be found in the WAP Characteristics table on www.forum.nokia.com

2.4.3 Optimize for size

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

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

2.4.4 Choose descriptive card titles

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

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

2.4.5 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 bulk of the information should reside there. 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 the card.

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

2.4.6 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 the total width or height.

2.4.7 Usage of do elements

Always use descriptive and short labels for all do elements. Favour local do elements instead of anchored links that do not fit into their surrounding context. Global do elements should be used only when especially needed at the end of the card. Always include a *prev* element in every card to enable backward navigation.

2.5 Pictures and tables

2.5.1 Avoid useless images

Downloading images takes time, and many users may switch off the loading of images for more speed. Try to optimise the size of 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 ** element) for images that convey information. Always use a null alternative text (alt="") for images that do not convey information, or are used for page layout or decorative purposes only.

2.5.2 Use reasonable table 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 table structuring.

2.6 Take care of backward navigation

Due to the differences between WAP and the Web in regard to the processing of the history, only the history processing described in the WML specifications is followed. This means that the service provider must be responsible for navigation as a whole.

2.7 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 item previously selected by the user should determine the header text. 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 header texts, and if the header text is too long, it is automatically truncated. Truncation is usually better than abbreviation, because the user might be confused by unfamiliar abbreviations that can be difficult to understand.

2.8 Performing a usability test

It is always good to perform a usability test of new applications. People who have not been involved in the design or development of the applications tend to notice potential usability problems that are 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 schedule does not allow for extensive testing.

3 INTRODUCTION TO THE USER INTERFACE

This chapter gives a short overview of the user interface type of Series 40 phones. Main focus is on the WAP services element (WAP browser) of the phone. The WAP browser user interface is designed to comply with the user interaction handling, known from other applications in the phone.

The user interface is a combination of specific user interface hardware like keys and display, user interaction conventions, which include input and output functions and their relationships, and audio-visual appearance, that is used in user interaction conventions. This chapter concentrates on the hardware element of the Series 40 user interface.

3.1 Nokia Series 40 phones

The Nokia browser in Series 40 phones has a WAP browser with enhanced features and increased usability. The deck size is 2.8kbit (including WSP headers) and cache size in terminal is 20kbit. Other WAP features include predictive text input in WAP (T9) and a picture viewer. Three different image formats are supported over the WAP; WBMP, static GIF and animated GIF. In addition, the June 2000 conformance (WAP1.2.1) phones support push service indication, WTAI public library and cookies. See the WAP Characteristics table in www.forum.nokia.com for more detailed information.


	Nokia 7210
	
WAP version	WAP1.2.1
Network	GSM900/1800/1900
Bearers	CSD, HSCSD, GPRS
Keys	Two soft key UI

Figure 3-1 Nokia Series 40 phones launched at the moment

3.2 User interface – display and keys

3.2.1 Two soft key display

The display is a high-resolution passive matrix colour display, 4096 colours, with a display resolution of 128 pixels (horizontal) and 128 pixels (vertical). The display consists of the application area, header area and the area used for the soft keys.

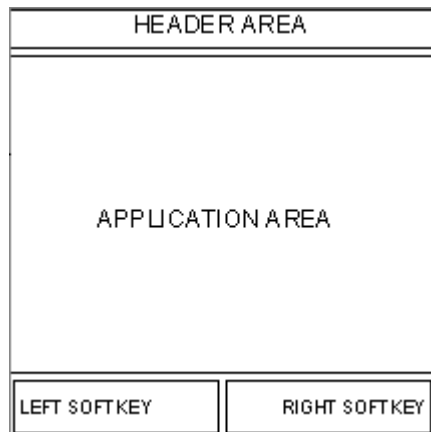


Figure 3-2 Structure of the browser screen

The application area is 122 pixels (horizontal) and 96 pixels (vertical), it includes 5 lines of main text and can also be used for displaying graphics, which is four ways scrollable (picture viewer) until the maximum deck size.

3.2.2 Keys in two soft key user interface

The **Send/Talk key** works as a selection key.

The **End key** exits a data call/GRPS connection and goes into idle state when pressed twice.

The **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 used as a yes/positive key. It contains options that execute commands and go deeper into the menu structure. Some example functions are Select, OK, and Options.

-The **right soft key** is used as a no/negative key. It contains options that cancel commands, delete text, and go backwards in the menu structure, such as Back, Exit, and Clear.

The **4 way scrolling/navigation key**

-**Scrolling up/down** allows the user to scroll the options, text and images in the current display up and down (vertical direction).

-**Scrolling left/right** allows the user to scroll the images in the current display in horizontal direction.

3.3 Display fonts

All fonts are proportional fonts. Proportional fonts give a dynamic and minimised width for each character, give improved readability of the display texts, and generally allow more characters to be displayed per line. On the other hand, having proportional fonts means that it is difficult to tell how many characters there can be in each line, cf. Characters 'W' and 'i'.

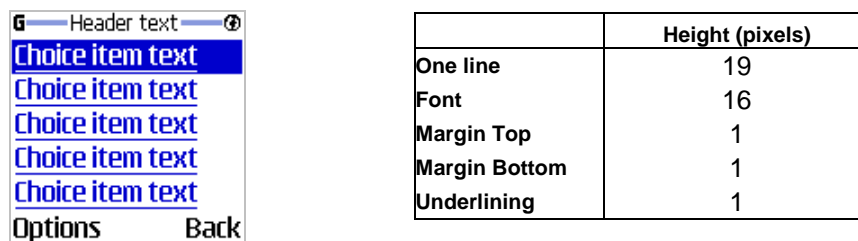


Figure 3-3 Information of one line

3.4 Accessing WAP browser

The Browser is accessed by selecting *Services* from the menu and entering a homepage defined for the currently active settings or a bookmark. For quick access to the browser, the user can make a long key press on the 0-key and the user is taken to the homepage that is defined for the currently active settings set. The access point and home page can be set in the Services -> Settings menu.

3.5 Navigation, labels and titles

The user should be provided with consistent card headers, element titles and do-element labels. The title of a <select> element is used in the header of a selection list or option groups list. The title of an <optgroup> element is used as an option groups 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 key.

3.6 Scrolling

The user can move up and down in card view when it contains elements (hotlinks, input elements, and selection list items) and static text areas between them. This is done by "jumping" from element to element. Scrolling through long static sections (that is, text lines) is done line by line. The first element of a card is highlighted by default if it is on the first display.

3.7 Picture viewer

The picture viewer offers the user a possibility to perform horizontal and vertical scrolling of pictures/animations that are too wide/high for the display. The picture viewer can only be activated by selecting the options list item *View Picture* if the currently active page contains at least one picture/animation. In the picture viewer, pictures/animations are displayed similarly to the way pictures/animations are displayed in the browser display except that only one picture/animation can be viewed at a time.

3.8 Cache

The browser places viewed cards in cache. Therefore, the previously visited cards can be quickly accessed, as there is no need to reload them. The cards stay in cache until it is manually cleared by the user or until the memory is full. See the WAP Characteristics table for information on cache size in different Nokia WAP phones.

4 WML ELEMENTS FOR THE NOKIA USER INTERFACE

This chapter is a guide for using WML 1.3 in designing services for Series 40 phones. It is an overview of general graphical user interface elements, that is, the WML elements that are critical from the point of view of rendering and the WML capabilities supported by the browser. All the elements introduced here are supported in every Series 40 Nokia phone introduced in this document. However, this chapter does not include all possible WML elements and attributes. WAP service developers should especially pay attention to sections 4.2 Table and 4.3 Images when optimising their existing services.

The WML elements are shortly described and demonstrated with the use of screenshots. The code examples include only the essential part in using the elements. i.e. they are not valid WML documents in itself.

4.1 Text formatting

The contents of a WML card are displayed in the card view. The card is a container of text, images, tables, and input elements. The order of elements in a card is significant, as they appear on the screen in the same order.

4.1.1 Paragraph and content alignment

The paragraph element, `<p>`, enables word wrapping and content alignment. A paragraph always starts on a new line. Normal static text information is always wrapped on a display, unless word wrapping is set off in browser settings.

Insignificant white space is ignored. If more than one white space separates the constituents of a text, these white spaces are ignored unless the `<pre>` element is used. See section 4.1.4 for more information on the `pre` element.

Content inside a paragraph can be aligned to the left, centre, or right by option; left alignment is displayed by default. Alignment is determined in paragraph attributes (for example, `<p align="right">`). Word wrapping can be set to wrap or nowrap by using the mode attribute in paragraph.

Example 4-1 Alignment `<align>`

```
<p align="center">
Align &quot;center&quot;
</p>
<p align="right">
Align &quot;right&quot;
</p>
<p align="right">

</p>
```

4.1.2 Line break

New lines in text can be defined by a line break. The line break element, `
`, can be used inside other elements as well, as long as it belongs to a paragraph. If a line break is defined after an `<a>` element, the line break is ignored. If an empty line is needed between two links, it is necessary to add two line breaks between the `<a>` elements.

4.1.3 Fieldset element

The fieldset element allows the grouping of related fields and text. The basic idea of grouping is to show the grouped items on the same display if possible. If the items cannot be displayed on the same display, the user is not given any notification. Fieldset always starts a new line and ends with a line break.

4.1.4 Pre element

The pre element is supported. When the pre element is used, the content shown is rendered as "pre-formatted" to the extent that is possible. This implies that white space is left intact when rendered, that the font in the cards remains the same as for plain text in a card, and that word wrapping is enabled/disabled according to how the browser is set.

4.2 Table

The `<table>` element is used together with the `<tr>` elements and `<td>` elements to create sets of rows and columns of data, such as text, images, and so on. It is possible to have text, images, and tables on the same card. The cells are shown in bordered rows and columns.

The size of the cells is dynamic, according to the content of the cells. The widest cells affect the width of the other cells in the same column. A minimum cell width of 10 pixels is necessary in order for the columns to remain discernible. The width of the rows will be adapted so that all can fit. Where more cells are defined than can fit on the screen, the cells are all of the defined minimum width. Where the content of the cells is too large to fit inside the cell, the content is truncated and three dots are added.

A maximum of one text, image or link element can be shown in each cell when no row in the table has been selected. Otherwise, text, image, as well as link elements, can be shown in a cell. In a text element, if line breaks are defined, several lines can be shown at a time. When no row has been selected, the longest line in a cell defines the width of the cell if the cell can be shown in full width. Otherwise, lines of excessive length are truncated and three dots are added. The full content of each cell can be viewed by scrolling.

The title of the table is shown when a row is selected. If the table has no title, the title of the card in which the table is situated is used as a title for the table.

Example 4-2 table element

<pre><table align="LCC" columns="3"></pre>		
<pre> <tr> <td>Date</td> <td>F'cast</td> <td>T °C</td> </tr></pre>		
<pre> <tr> <td> <anchor title="date">M 6/7 <go href="#date" method="get" sendreferer="false"/> </anchor> </td> <td> <anchor> <go href="#descr" method="get" sendreferer="false"/> </anchor> </td> <td>25°C</td> </tr></pre>		
<pre> <tr> <td>T 6/8</td> <td> </td> <td>27°C</td> </tr></pre>		
<pre> <tr> <td>W 6/9</td> <td> </td> <td>24°C</td> </tr></pre>		
<pre> <tr> <td>T 6/10</td> <td> </td> <td>28°C</td> </tr></pre>		
<pre> <tr> <td>F 6/11</td> <td> </td> <td>29°C</td> </tr></pre>		
<pre></table></pre>		

4.3 Images

The graphic formats Wireless Bitmap (WBMP), static GIF (*.gif 87a), animated GIF (*.gif 89a) are supported. The static graphics area is a maximum of 122 pixels wide and 96 pixels high, but vertically and horizontally scrollable up to the maximum deck size (2.8kbit). The animated GIF graphics area is a same than static graphics area. Images fitting in the graphics area are centred unless specified otherwise by the service provider.

However, displaying images is optional and can be turned off in browser appearance settings. Images can also be used inside a table and as a link (see example 4-2 above). An image that functions as a link is represented with a full-line frame around it. There cannot be text next to an image; the image will always start its own line.



Figure 4-1 Image and header text

If a requested image does not exist or could not be fetched, a small default picture will be shown. It is also recommended to use the image alt attribute value as an alternative textual representation for the image.

Example 4-3 img element

```

```

4.3.1 Animated GIF

The most recent Nokia browsers¹ support animated GIF (*.gif 89a) image format. The maximum number of frames allowed is 13 (note that the maximum file size is 2.4 kb).

¹ All phones that were introduced in this document support GIF image formats in WAP browser

4.4 Links

The anchored link or <a> element specifies a hotlink. Links are underlined and coloured as blue. When a user selects a link, an associated task is performed. When the link is active, the options list of the browser display is extended with the option "Open link". The user can also select a link by pressing the Send key.

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

A hotlink is automatically followed by a line break. In case a line break is defined after an <a> element, the line break is ignored. If an empty line is needed between two links, it is necessary to add two line breaks between the <a> elements. If a hotlink does not have a title or label, the URL address of the link is displayed in its full length, and functions as a normal hotlink.

Example 4-4 Image as a link

```
<anchor>text<go href="#card2"/>
</anchor>

<anchor>

<go href="#card2"/>
</anchor>
```

4.5 Do element

The `do` element provides a general mechanism for the user to act upon the current card. The `do` elements are mapped behind the browsers options list together with the preloaded options. If a `do` element contains a `prev` task, the right soft key is labelled "Back" and the options list is extended with "Back". Use of a `prev` task is highly recommended in every card because it enables backward navigation. Do elements have a label that is the text of the item in the browser's options list. If the label is available it is used as an option list item. If there is no label, a default text is displayed, according to Table 4-1 below. However, it is highly recommended to use the `label` attribute.

Type	Static text
Accept	OK
Prev	Back
Help	Help
Reset	Reset
Options	Options
Delete	Delete
Unknown	Unknown

Table 4-1 Do elements label default text

4.5.1 Prev task

The author of the WAP service is always responsible for the working navigation model within its own service. If there is a need for back functionality, it should be explicitly defined with the `<prev>` type `<do>` element. The WAP browser does not have any implicit back functionality.

4.5.2 Card-wide and deck-wide scope

The `do` element can be expressed in two ways: in card-wide scope or in deck-wide scope in a template. `Do` elements will be stored in the browser's options list together with the preloaded options.

Example 4-5 Deck-wide backward navigation, `prev`

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

Example 4-6 Card-wide backward navigation

```
<do type="prev" label="Back">
<prev/>
</do>
```

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>

```

4.6 WML input processing

Input processing enables the user to input requested information to the service. There are two kinds of input element; text fields determined by an input element and selection lists determined by a <select> element. <Option> elements are used to specify a single choice option in a <select> element. <Option> elements can be grouped by using an <optgroup> element (see example 4-9).

4.6.1 Input elements

The `input` element specifies a character entry object. The input element is displayed with brackets. By selecting the input element, the user evokes the editor. The editor is never displayed directly in card view. The editor layout is alphanumeric data query layout. The options list of the browser display is then extended with the "Edit" option. The T9 option is enabled in the browser. The user can select the input element by pressing the Send key. The input element consists of a title, brackets, and a value. There can be a default value inside the brackets. The title of the element is only displayed when the editor box is displayed, which occurs after the input element has been selected. The value with the brackets is always wrapped to the next line. If the value cannot fit into one line, the end of the content is truncated and three dots are added.

Direct input access

The user can also select the input element by pressing any number key when the edit box appears and the pressed key is already placed in the edit box. In other words, the user can just start inputting characters and the edit box appears automatically.

When T9 (predictive text input) is activated, the soft keys are "Options" and "Back" when the editor is empty. Otherwise, the soft keys are "Options" and "Clear". This also applies if T9 is deactivated. Input fields can be specified by the input element 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>

```

4.6.2 Select element

Selection lists are elements that specify a list of options for the user to choose from. Two kinds of selection list are supported; single choice and multiple-choice lists. The user can select multiple choices, if the `multiple` attributes are set as `true` (selected items are marked with a selection symbol). The user can highlight and select the selection list item on a card. The options list of the browser display is extended with "Select". The user can also select the selection list by pressing the Send key. That same item will be highlighted and displayed with a selected icon in front of it, once the user re-enters the list during the same browsing session.

A selection list item on a card looks similar to an `input` element item; it consists of a title, brackets, and a value. There is always a default value inside the brackets in a single selection list (in a multiple selection list there are, by default, three dots between the brackets unless specified otherwise by the service provider). If there are no options in the selection list, three dots are displayed between the brackets. The title is not displayed in the card itself. The value with the brackets is always wrapped to the next line, which means that there will never be text and a value with brackets on the same line. If the value cannot fit onto one line, the remaining part of the content is truncated and replaced with three dots. The title of the selection list is the first item of the card containing the selection list. If the value is a multiple selection value, commas separate the selections. If the value cannot fit into one line, the end of the content is truncated and replaced with three dots. The user can view the value between brackets in full by pressing the # key. Until the # key is released, the content that can fit on the screen is viewed. The soft keys are hidden while the

content is viewed in full. The layout in full view is the same as the one selected by the user for browsing.

The `<option>` element specifies a single choice option in a select element. Options can be grouped with an `<optgroup>` element, which specifies a group of choice items in a select element. An `<optgroup>` element is displayed with an icon in front of the title. An `<optgroup>` element can be used in single selection lists as well as in multiple selection lists. Only in multiple selection lists is it possible to have option groups that allow multiple selections.

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 8310">
          8310
        </option>
        <option value="Nokia 5510">
          5510
        </option>
        <option value="Nokia 6310">
          6310
        </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>
```

5 OPTIONAL WAP 1.2.1 FEATURES

This chapter is an introduction to the optional WAP 1.2.1 features specified by WAP Forum. These features are supported in most WAP 1.2.1 compliant phones.

5.1 Push Service Indication

Push service indications are an ideal way to activate users. They can be used to notify users of updated content or alert in their areas of interest. The service provider can push a service indication content type to the phone display, which is then saved to the Service Inbox in Service menu. The user can view indicated content afterwards by using the Service Inbox. The service indication should contain at least a text with a URL address indicating a service.

5.1.1 Push indication structure

Service indication contains text that is displayed to the user, and a URL of content which may be dynamically generated for each push message in the application server.

Messages that have already been sent can be replaced and deleted. Messages with the same message id can also be put into the correct order by creating time. Service providers can also determine the user attention priority level (none, low, medium, high).

5.1.2 Service Inbox

Push service indications are saved in the Service Inbox. The Service Inbox can be accessed through the options list under Services and under the options list belonging to the browser display. In Service Inbox, the received service indications can be viewed, deleted or downloaded. Note that receiving push services is set to off by default; it can be set to on in Service inbox settings.

5.2 WTAI public library – Wireless Telephony Applications Interface public library

Three WTAI (Wireless Telephony Applications Interface) public library functions are supported. You can make a call while browsing, send DTMF tones, and save numbers and names to the phone book. These functions, MakeCall, SendDTMF and AddPBEntry, can be used anywhere in the WML code in the same way as when URI or script references are used for normal non-WTAI references.

5.2.1 Making a phone call

The MakeCall function is used for making a phone call directly from the WML browser application. A typical example of the usage of MakeCall is when the user selects an anchored link, which initiates a phone call. When a MakeCall function is executed, the user is asked to confirm that the dial-up is to be made. If the phone number that is about to be called does not exist in the phone book, a confirmation query with the phone number is displayed. If the number to be called exists in the phone book, a confirmation query is displayed with the name fetched from the phone book. If the user accepts to make the phone call, another confirmation query is displayed with the text "Quit browsing and make the call?"

If the user accepts both queries, the browser session will be terminated and the browser application will be closed down. The call will then be initiated. When the

call is terminated, the phone is in idle state. If the user rejects the latter confirmation query, the call will be initiated from within the browser display and the user can browse cached content during the call. No in-call options such as Mute, Hold etc. are accessible from within the browser application. The options list belonging to the browser display remains unaffected by the fact that a call is active.

5.2.2 Send DTMF tone

The user can send *DTMF tones* through the network by executing a WTAI function called SendDTMF. Just like MakeCall, a typical way of using SendDTMF is to assign the SendDTMF function to an anchored link. The user must have a voice call active. When the DTMF tone has been sent, the user returns to the card/deck where s/he initiated the sending. The voice call remains active.

5.2.3 Add a phone book entry

AddPBEntry-function offers an easy way of storing a phone number and a corresponding name onto the phone book application from a WML card. The service provider provides the name and number. When the user activates an AddPBEntry function and the service provider has declared a name and a number, the phone displays a confirmation query with the text "Save name." When the user accepts the confirmation query, another confirmation query with the text "With number +44 123 123 4567" is displayed and the number is stored in the phone book application. When saving is done, the currently active card is displayed again. Note that the user does not have to be online or have any voice calls active in order to use the AddPBEntry function.

5.3 Accesskey

Service providers have the possibility of using the accesskey attribute of the `a` element and of the anchor element. This permits the user to open links with a long press of the associated access keys. The allowed keys for access key use are the 0-9 number keys. The service provider should indicate to the user which keys to press in order to open the desired links. The key can be indicated to the user by adding the label of the key to the link concerned, e.g. if the service provider associates the "2" key to a link, the service provider should add the number "2" to the label of the link.

Example 5-1 Using accesskey "2"

```
<a accesskey="2" href="phone.wml">(2)a </a>
```

Only one link can be associated with each accesskey. A key will also open its corresponding link if the service provider has assigned it as an accesskey but does not indicate this to the user. If the user tries to open a link by using a key that the service provider has not associated to any link, the key used does not react to the action of the user. An accesskey has an effect on a link even if the link is not highlighted. The link chosen by the user is highlighted after the corresponding accesskey has been given a long press. This is visible while the terminal is fetching the content of the link. A long press of an accesskey also has an effect when a card contains both accesskeys and a highlighted input element. An input

element can be activated by an associated accesskey through a long press in the same way as anchored links as described above.

5.4 Cookies

Cookies enable the storage of data like user information to ease the browsing process for the user, by reducing the amount of information input required. For example, login can be configured dynamically so that it is displayed only if the application cannot identify the user.

5.5 Content download over WAP

Content download is a technology that is used to deliver digital content such as entertainment and business applications to mobile terminals. Another important application is for personalizing terminals according to a user's preferences and lifestyle. Content downloading can be initiated by an end user or by a network application.

The content download mechanism between the mobile terminal and content source is based on open technologies such as HTTP (and WSP) and Java Community Process technology. This technology allows the creation of an open, multivendor market for terminal vendors as well as for network server and broker vendors.

For more information about the Content Download, please go to <http://www.forum.nokia.com/>

A. WIRELESS APPLICATION PROTOCOL WAP

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. The WAP Forum maintains these protocols and their related standards and specifications.

The WAP Forum consists of a number of high-tech companies from the information technology, software, and telecommunication industries. The objectives of the WAP Forum are to:

- Bring Internet content and advanced data applications to digital mobile 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

WAP architecture

WAP and Internet architecture

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

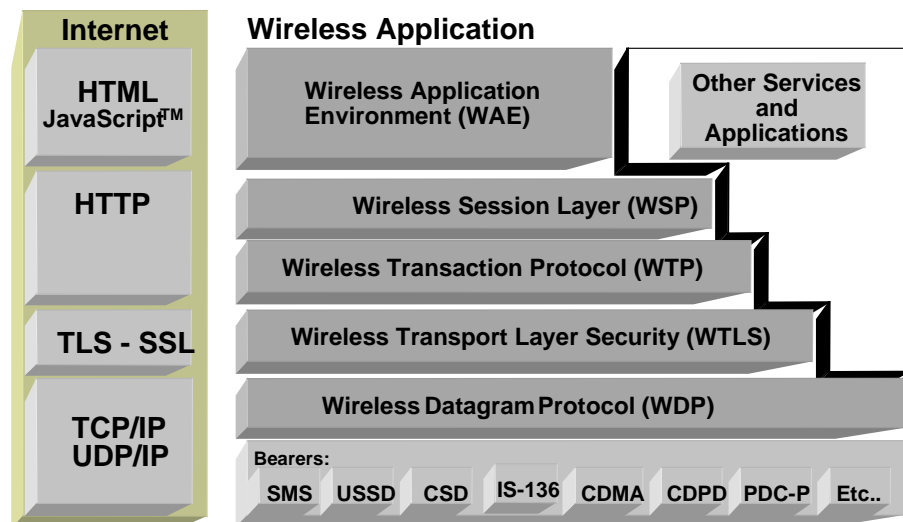


Figure 1 Comparison of Internet and WAP Architecture

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

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. WAE contains support for the following functionality:

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

Wireless Session Protocol (WSP)

The Wireless Session Protocol (WSP) provides the application layer of 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 optimised for low bandwidth bearer networks with long latency.

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.

Wireless Transport Layer Security (WTLS)

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

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 Wireless Markup Language (WML) and WMLScript, and stored on either a normal Web server (origin server) or directly on the WAP gateway. The content stored on the Web server is accessible from 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 Internet or Intranet. The data sent between the origin server and the handset is binary encoded to optimise 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 minimise network load.

Figure 2 shows the network-related elements required for developing and offering services to mobile users.

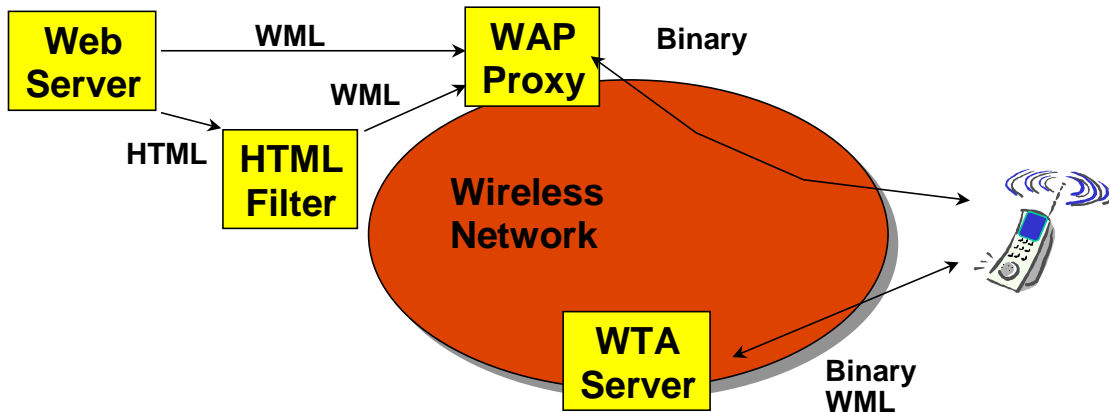


Figure 2

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 Internet or Intranet. It encodes the information which is to be sent to the mobile terminal in binary format 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.

Further information

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

B. LIST OF TERMS AND ABBREVIATIONS

Term or abbreviation	Description
Card	A single WML unit of navigation and user interface.
Deck	A collection of WML cards that is the smallest downloadable 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 create large applications in a single deck.
Do element	A way of defining the binding between events (such as a user selection) and a task (for example, to go somewhere).
DTMF tone	Standard tone used to transmit numeric data via telephone. Generated when number keys are pressed.
GPRS (General Packet Radio Service)	Mobile service, which gives a packet switched access over GSM to external data networks with high-peak transfer capacity.
Hyperlink	A link within a document or card that allows quick navigation to another document (card).
Input element	A markup element that allows for interaction with the user, that is, letting the user input textual values. These can be used for inputs to locally stored scripts, or for parsing to origin servers.
Multiple selection list	A selection list, which allows the user to scroll and select several (or none) options to be active.
Navigation	The concept of moving between menu items, decks, and cards.
Proportional font	A font in which different letter widths do not necessarily have the same number of pixels. 'W' and 'l' are good examples.
Selection list	List of user-selectable options.
Soft key	The phone has one or more function keys whose function depends on the state of the phone. These keys are called soft keys because there is an associated text on the display.
T9 in WAP	Predictive text input in browser input fields.
Word wrapping	Formatting rule for text appearance. Specified in Editing (UI components).
WML (Wireless Markup Language)	Markup language based on Extensible Markup Language (XML), which is intended for use in specifying content and user interface for narrowband devices, including mobile phones and pagers.
WTAI	Feature that provides the means to create telephony applications by using a WAE user-agent with the appropriate WTAI function libraries.

Free Manuals Download Website

<http://myh66.com>

<http://usermanuals.us>

<http://www.somanuals.com>

<http://www.4manuals.cc>

<http://www.manual-lib.com>

<http://www.404manual.com>

<http://www.luxmanual.com>

<http://aubethermostatmanual.com>

Golf course search by state

<http://golfingnear.com>

Email search by domain

<http://emailbydomain.com>

Auto manuals search

<http://auto.somanuals.com>

TV manuals search

<http://tv.somanuals.com>