

WapIDE 3.1

User's Guide



Ericsson is the leading provider in the new telecom world, with communication solutions that combine telecom and datacom technologies with freedom of mobility for the user. With more than 100,000 employees in 140 countries, Ericsson simplifies communications for its customer – network operator, service providers, enterprises and consumers – the world over.

WAP™ and WAP Forum™ are registered trademarks of Wireless Application Protocol Forum, Ltd., a registered company in the United Kingdom.

Java™ and JDK™ are registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Microsoft®, Windows® and Windows NT® are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe® and Acrobat® Reader™ are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

First edition (April 2001)

This document is published by Ericsson Radio Systems AB, without any warranty. Improvements and changes to this text necessitated by typographical errors, inaccuracies of current information or improvements to programs and/or equipment, may be made by Ericsson Radio Systems AB at any time and without notice. Such changes will, however, be incorporated into new editions of this document. Reproduction, distribution or transmission of part of or all of this document in any form without the prior written permission from Ericsson Radio Systems AB is prohibited.

All rights reserved.

© Ericsson Internet Applications AB, 2001

Contents

Introduction	6
Purpose and scope of the user's guide	6
Who is this guide for?	6
Related documents and other resources.....	6
Related Internet sites.....	6
Related documents	6
Changes from previous version	7
Support	7
Typographical conventions	7
System overview	8
Ericsson WapIDE	8
Browser	8
Application designer.....	9
Push initiator.....	9
Installing WapIDE	10
System requirements	10
Software requirements	10
Hardware requirements.....	10
Installation procedure	11
Installation overview	11
WapIDE files.....	11
Uninstalling WapIDE	11
Using the WapIDE browser	12
Starting the browser.....	12
Browser window.....	13
Accessing an application.....	14
Selecting the device	14
Using the R320s/R520m browser.....	16
Using the device buttons.....	16
Entering data	16
Using the options menu	17
Using the R380s browser.....	18
Navigating.....	18
Entering data	18
Using the buttons	19
Working with bookmarks	19
Using the browser trace	20
Setting browser preferences.....	21
Browser settings.....	21
Gateway settings.....	22
Cache settings	23
Device-dependent settings	24
Testing applications in other character sets	25
Standard texts.....	25
WML presentation	25

East Asian text input	25
Using the WapIDE application designer	26
Starting the application designer	26
Application designer window	26
Working with projects	28
Creating a new project	28
Opening an existing project	28
Working with files	29
Creating new files	29
Adding files to a project	29
Removing files from a project	29
Saving and closing files	30
Editing WML decks	31
Structure of WML decks	31
Inserting elements in a WML deck	31
Formatting text	32
Validating a WML deck	32
Encoding a WML deck	33
Editing WMLScript files	33
Testing applications	34
Creating applications in other character sets	34
Setting application designer preferences	36
Fonts	36
Colors	36
Code style	37
Working with push messages	38
WapIDE implementation of push	39
Using the push initiator	40
Push settings	42
Push operation	42
Content definition	45
PPG communication	49
Receiving push messages	49
Service indication	49
Other message types	50
Pushing messages to a real terminal	50
Writing applications that generate push messages	50
Setting up a WAP application environment	51
Overview	51
Accessing local WML files	51
Using a gateway on the Internet	51
Using a local gateway	52
Configuring a web server	53
Creating dynamic WML applications	53
Managing sessions	53
Creating WMLScript applications	54
Installing a local gateway	55
Downloading and installing the gateway	55
Configuring the gateway for push	55
Define push initiator	55
Turn off authentication	56

Define subscriber	57
Define receiver IP address	57
Appendix A: Device limitations	59
Devices supported	59
Optional WAP features not supported	59
Functions not supported by WapIDE	60
Glossary of terms	61

Introduction

Purpose and scope of the user's guide

This document provides information on how to install and use Ericsson's WAP Integrated Development Environment (WapIDE) 3.1. WapIDE is a Software Development Kit (SDK) that helps you develop and test services for a WAP platform.

Who is this guide for?

This user's guide is for developers of WAP services. It assumes that the user has a basic knowledge of WAP. Otherwise we refer to the WAP resources listed in *Related documents and other resources*.

Related documents and other resources

Related Internet sites

Ericsson	http://www.ericsson.com
Ericsson Developers' Zone	http://www.ericsson.com/developerszone
Ericsson Mobile Internet	http://www.mobileinternet.ericsson.com
WAP Forum	http://www.wapforum.org

Related documents

These related documents can be found at <http://www.ericsson.com/developerszone>:

Mobile Phone R320, Design Guidelines for WAP Services

Mobile Phone R380, Design Guidelines for WAP Services

Mobile Phones R520, T20 and A2628, Developers' Guidelines, WAP Services

Changes from previous version

The main differences between WapIDE 3.1 and 3.0 are:

- An application designer (WML editor) is added.
- Support for the R380s smartphone is added.
- Support for the WAP push framework is added.
- Support for connection-oriented mode is added.
- Support for user agent profiles (UAProf) is added.
- The WML encoder now supports other character sets.
- WapIDE now uses the same cache options as the real terminals.
- Devices can be shown in different colors.

Support

Support can be found at the Ericsson Developers' Zone (see *Related Internet sites*) free of charge.

Typographical conventions

The following typographical conventions are used in this document:

Bold	Names of commands in menus, buttons
<i>Italic</i>	Specific terminology
<code>Courier</code>	Computer text, file names

System overview

Ericsson WapIDE

WapIDE is a Software Development Kit (SDK) that enables operators, application developers, or any interested party to develop and test real WAP applications swiftly and easily. WapIDE can be downloaded free of charge from the Ericsson Developers' Zone.

The main functions in WapIDE are:

- The *browser* simulates a WAP device and allows you to test WAP applications on different Ericsson phones.
- The *application designer* lets you create and test your own WAP applications.
- The *push initiator* sends push messages to the WapIDE browser or a real terminal.

An online version of this *User's Guide* can be accessed directly from the Windows Start menu or from inside WapIDE by using the help menu.

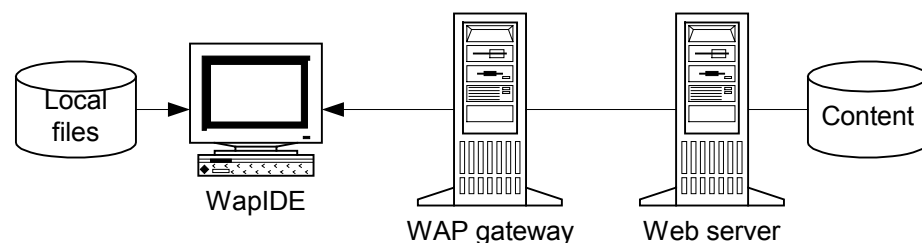
Browser

The WapIDE browser allows the user to access WML decks and cards using a simulated WAP device. The following Ericsson devices are currently supported:

- R320s
- R380s
- R520m

The Chinese versions of these terminals are also indirectly supported since Chinese characters can be entered from the computer keyboard.

The browser can access content from a web server via a *WAP gateway* or from a local disk.

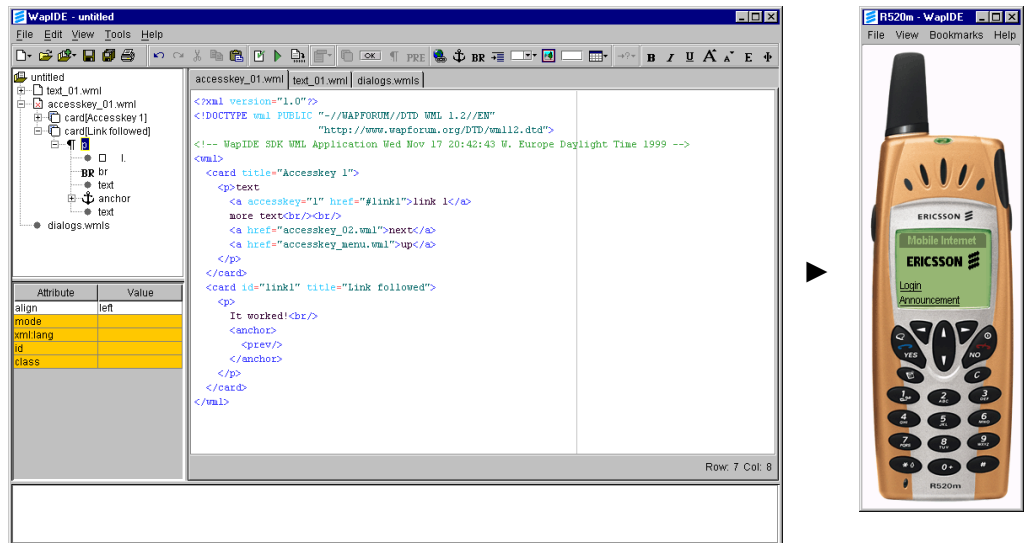


Application designer

The WapIDE application designer is a WML editor with which you can design and test WAP applications. There is also a WMLScript editor for writing and compiling WMLScript code.

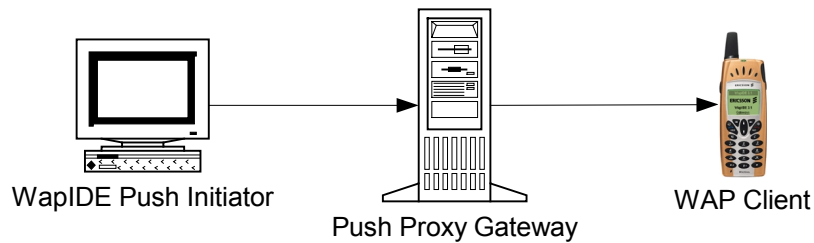
WML, WMLScript, and other files can be managed in projects.

The application designer is integrated with the WapIDE browser so you can easily test your applications on different devices.



Push initiator

The WapIDE push initiator is used to create and send push messages to the WapIDE browser or a real terminal.



Installing WapIDE

This chapter describes the system requirements and installation procedure for WapIDE.

WapIDE can be found at the Ericsson Developers' Zone (*WAP -> Developers' tools*). To access the zone you have to register at the web site. The registration gives you access to the documentation and developer resources located at the site.

System requirements

Software requirements

- Microsoft Windows NT 4.0, Windows 98, or Windows 2000.
- Java 2 Platform, version 1.3.0 or later. Java can be downloaded from <http://java.sun.com>.
- Microsoft Internet Explorer 5, or a later version, is required for local WML file access, including the use of the application designer.
- The font Arial Narrow is required to get a correct simulation of the real telephones. Arial Narrow is part of many Microsoft software packages such as Office 97, Outlook 2000, Word 2000, etc.
- Adobe Acrobat Reader 3.0 or later is required to view online documentation. Acrobat Reader can be downloaded from <http://www.adobe.com>.

In some cases it is also necessary to install a WAP gateway as described in *Setting up a WAP application environment* on page 51 and *Installing a local gateway* on page 55.

To use WapIDE with other character sets, such as Chinese, the following is recommended (see *Testing applications in other character sets* on page 25 for more information):

- Windows 2000.
- Java 2 Runtime Environment (JRE) 1.3.0, internationalized version.

Hardware requirements

- 20 Mbytes free disk space is required for installation.
- Pentium II, 266 MHz and 128 Mbytes RAM is recommended to get acceptable performance.

Installation procedure

To install WapIDE on Windows:

1. Install the prerequisite software listed above.
2. Download the file `wapIDE_31.exe` from <http://www.ericsson.com/developerszone>.
3. Run the file `wapIDE_31.exe` (by double clicking on it) and follow the instructions.

Installation overview

By default, an **Ericsson WapIDE 3.1** menu is added to the **Programs** section of the Windows Start menu. The structure is:

Ericsson WapIDE 3.1 ▶	Application Designer	(The WML editor)
	Browser	(The WAP browser)
	Push Initiator	(The push message creator)
	Readme	(Release notes)
	User's Guide	(This manual in PDF format)

WapIDE files

The default directory for installation is
`C:\Program Files\Ericsson\WapIDE 3.1`

The following subdirectories are created:

WapIDE 3.1	Root directory.
bin	Java archive (JAR) files and DLLs.
dat	WapIDE settings, bookmarks, and DTD files.
device	Device-specific code and settings.
doc	This User's Guide in PDF format.
samples	Contains samples, a default home page, and is the default root directory for <code>file://</code> accesses.

Uninstalling WapIDE

To uninstall the WapIDE use the Windows **Add/Remove** Program in the Control Panel folder in the **Settings** section of the Windows Start menu.

Note: Deleting the WapIDE files and directories manually will not completely uninstall it. You have to use the procedure above.

Using the WapIDE browser

The WapIDE browser is used to view WAP applications. It can be used instead of a WAP device to access WML decks developed by you or others. It also interprets WMLScript.

The browser can simulate different devices. It also supports applications in languages with different character sets, such as Chinese.

There are two ways to load content to the browser; from a:

- Local WML file
- Web server via a WAP gateway.

To access web servers on the Internet, you can use a WAP gateway on the Internet. If you want to test applications on local web servers, or if you are placed behind a firewall that prevents you from communicating with external gateways, you can download a gateway from the Ericsson Developers' Zone. Refer to *Setting up a WAP application environment* on page 51 for more information.

Starting the browser

Start the browser by selecting

Programs ► Ericsson WapIDE 3.1 ► Browser

from the Windows Start menu.

The window shown on the next page is opened. Initially, the browser starts with the R520 device and the default home page (a local WML file).

The default gateway is one that is available for external test use at the Ericsson Developers' Zone. To change the gateway, refer to *Gateway settings* on page 22.

Browser window



At the top of the browser window is a menu bar, a toolbar and a location bar. Below that is the device.

The toolbar offers quick access to common menu choices. It also contains a “progress indicator” that rotates while a file is being loaded. The location bar allows you to enter a URL to load, and also keeps an history list of previously entered URLs. The toolbar and location bar can be removed (using the **View** menu).

Initially, the window is shown with a default size, but it can be resized (e.g. to make the location field wider). The device picture can be moved (by dragging it), centered, or locked within the window (using the **View** menu or a pop-up menu).

Accessing an application

Applications are accessed by an URL. To load a URL, you can do one of the following:

- Type the address in the Location field as a normal Internet URL, e.g.:
<http://mobileinternet.ericsson.com>

You can also load a local WML-file by typing
`file://C:/path/file.wml`

If you use a relative file URL that does not start with e.g. C:, the reference is relative to the WapIDE `samples` directory. For example,

`file://project1/test.wml`

is the same as

`file://C:/Program Files/Ericsson/WapIDE 3.1/samples/
project1/test.wml`

- Select an URL from the history list for the Location field.
- Select a bookmark as described in *Working with bookmarks* on page 19.
- Select **Load URL** from the **File** menu and type a URL in the same way as above.



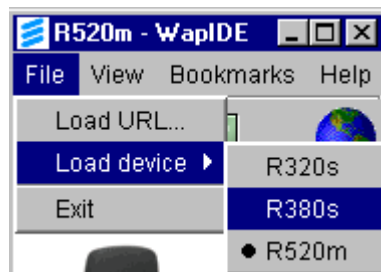
Pressing the **Browse** button opens a file chooser window where you can select local WML files to load (`file://...` access).

The browser will load the application and show the first card in the display.

Note: At Ericsson Mobile Internet there is a list of links to WAP-enabled sites that you can try.

Selecting the device

To open the current card with another device, select **Load device** from the **File** menu.



The window switches to the selected device.

R320s



R520m



R380s



The browser capabilities of the different devices are described in the following sections.





For further information on the user interface of Ericsson WAP devices you can download *Design guidelines* for different Ericsson WAP terminals from the Ericsson Developers' Zone.

Using the R320s/R520m browser

The R320s and R520m devices are very similar.

Using the device buttons

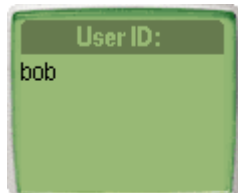
When the browser is started you can see the application in the display and use the buttons on the device to enter commands and data.

Button	User action	Device action
	Short click	Follows the current link, confirms input or selection, etc
	Hold-down click	Shows a list of options, as described in <i>Using the options menu</i> below
	Short click	Goes back in history
	Up / Down	Scrolls up or down in the current card or selection list
	Left / Right	Scroll left or right in tables (or between multiple links on the same row)
	Short click	Text entry in input fields. The * and # buttons are also used for selecting check boxes. For cards with “access keys”, these buttons can also be used as a quick way to activate a link or input field (R520m only). Access keys are indicated with a small icon next to the link or input field.

Entering data

You can enter data in input fields in two ways:

- By clicking on the buttons on the device. This works in the same as for the real telephones (e.g. press button **2** three times to get the character C).
- By using the computer keyboard. This is a faster way to enter text. The **Enter** key corresponds to the **Yes** button and the **Esc** key to the **No** button.



Input fields can be in different formats, such as numeric, alphabetic, etc and of different lengths. WapIDE verifies that the input is in the correct format, and stops

you from entering e.g. alphabetic characters in numeric fields or strings that are too long.

When you enter text on a real telephone, a character menu is shown in the upper right corner each time you press a button. WapIDE shows similar information in a small pop-up window (“tool tip”) when you hold the mouse over a device button.

Using the options menu

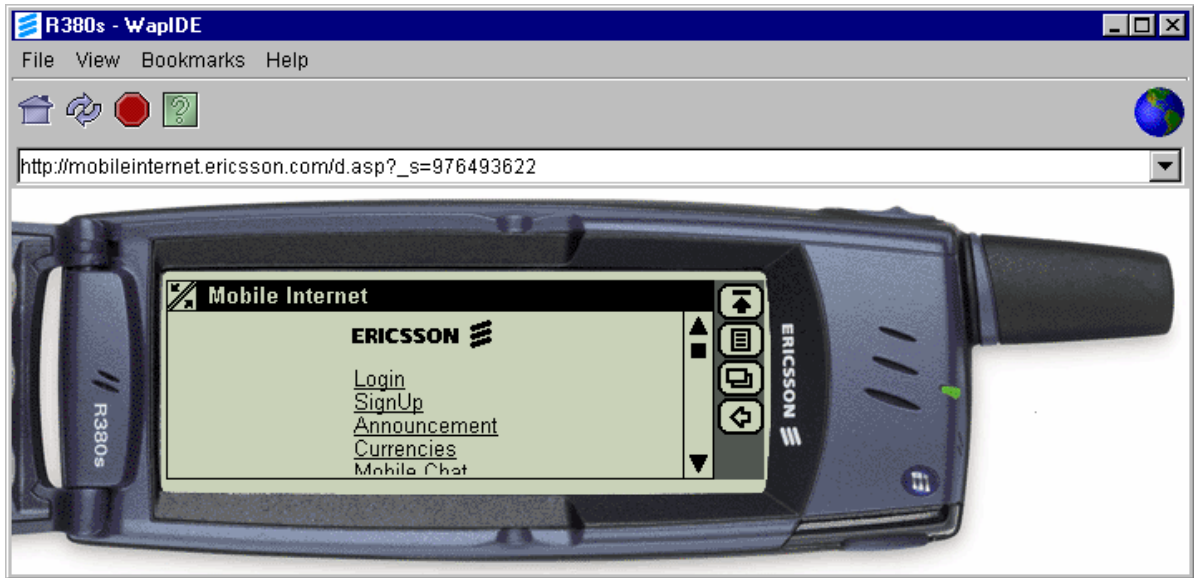
When you hold down the **Yes** button, an options menu is shown. (On the R320s device, a short click on **Yes** will also display the options menu if no link is in focus. On the R520m device, the microphone button will also display the options menu.) The options menu contains any actions defined in the current card (“soft options”). In the example below, `prev` and `TV4.SE` are such options.



The options menu also contains a number of fixed options (such as `1 Mobile Internet` above) but these options have no effect in WapIDE. The same functions (reloading the page, adding a bookmark, etc) can be performed from the menu bar.

Using the R380s browser

The R380s device is quite different from the traditional mobile phones. It provides a large touch-screen and is controlled with a pen instead of buttons on the device.



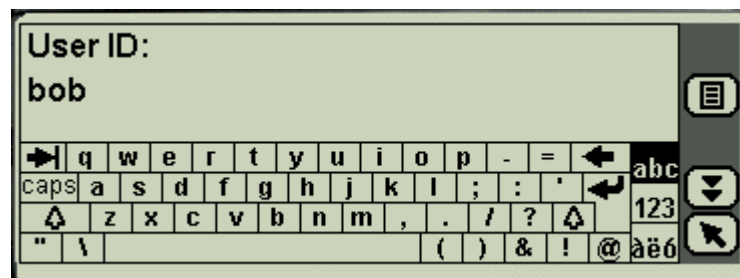
Navigating

Follow links by clicking on them with the mouse (like in a web browser). Click on the arrows in the scroll bars to scroll horizontally or vertically.


The browser keeps a history list of visited applications. The **Back** button goes back to the previous card. When the first card in the current application is reached, **Back** goes to the first card in the previous application.

Entering data

Open input fields and single-select fields by clicking between the $\langle \rangle$ brackets.








WapIDE allows text input from the computer keyboard only. The real R380s device allows text input from an on-screen keyboard or a character recognition screen but these methods are not supported by WapIDE. The on-screen keyboard is displayed but has no effect.

Click on  to confirm the input and return to the main window.

Using the buttons

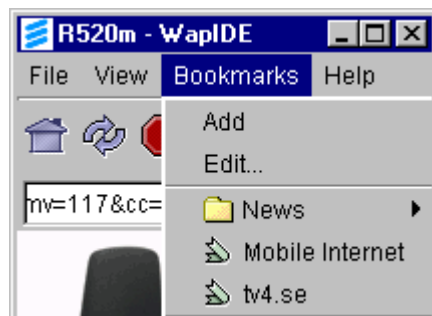
At the right-hand side of the screen is a menu with buttons:

-  The **desktop** button does nothing (WapIDE only emulates the WAP browser part of the R380s).
-  The **tasks** button opens a menu with different options but selecting them has no effect. Some of the functions on this menu (reloading the page, adding a bookmark, etc) can be performed from the WapIDE menu bar.
-  The **view** button works in the same way as the tasks button.
-  The **back** button goes back to the previous page as described above.
-  The **stop** button is only shown while a page is being loaded. It works in the same way as the stop button on the WapIDE toolbar.

Working with bookmarks

Bookmarks work in the same way as in a web browser. Add a bookmark for the current card by selecting **Add** from the **Bookmarks** menu. It is recommended that you bookmark only the first card in an application. Other cards may have dependencies on the browser context.

To retrieve a bookmarked card you select it from the **Bookmarks** menu.

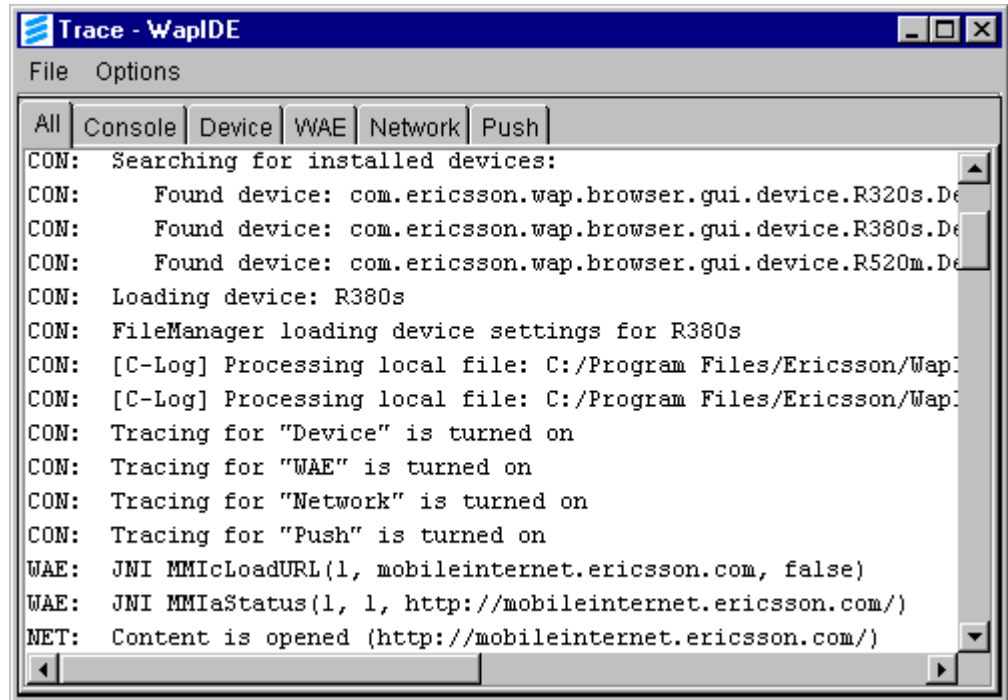


When you add a bookmark, a file with the bookmark name is created. As shown in the example above, you can create a tree structure of bookmarks by moving them to subfolders. This must be done outside WapIDE using e.g. Windows Explorer.

Bookmarks-Edit... opens a window where you can delete and rename bookmarks.

Using the browser trace

In the browser trace you can see system events of different types. Selecting **Trace** from the **View** menu opens the trace window. The trace window is also opened automatically if an error (such as a timeout) occurs.



The trace messages are divided into different tabs:

- All* All trace messages in the order they were written.
- Console* Startup messages, settings, etc. The console trace is always on since it can contain startup errors and other important information.
- Device* Device-related traces.
- WAE* Wireless Application Environment (WAE) information, related to the WAP stack and interpretation of WML code.
- Network* Network communication traces such as files read from the network.
- Push* Push-related traces.

By default, only the Console trace is active. The other traces (Device, WAE, Network, and Push) can be activated using the **Options** menu.

Using **File-Clear**, you can clear the traces written so far for the current tab. **File-Clear all** clears all traces.

Some of the messages in the trace window contain internal WapIDE information that is normally not relevant to the user but can be important if you report a WapIDE problem.

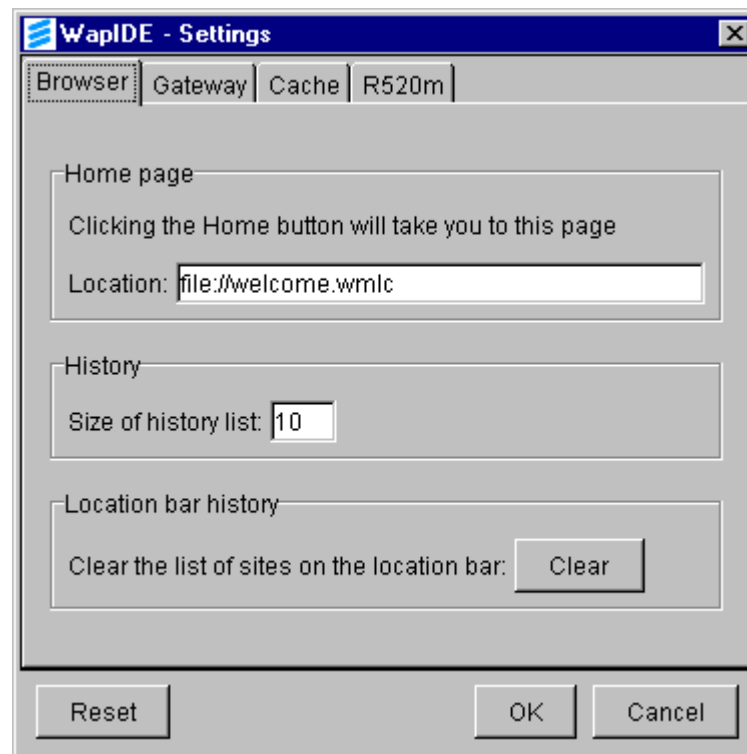
Setting browser preferences

Selecting **Settings** from the **View** menu opens the settings window. There are four tabs in the window:

- Browser
- Gateway
- Cache
- Device-dependent

Browser settings

This tab contains general browser settings.



Home page

The home page is loaded when WapIDE is started and when the **Home** button is pressed.

History

WapIDE keeps an internal history list of loaded WML decks. The size of this list limits the number of times you can go back to the previous card.

Location bar history

The location bar history list contains all the URLs entered in the location field. Pressing **Clear** empties this list.

Gateway settings



Gateway

Select the WAP gateway to use from the list. The default gateway is one that Ericsson provides for external test use (IP address 195.58.110.201). See the Ericsson Developers' Zone for more information.

Add...

When you add or edit a gateway, the following information can be entered:

Edit...

- *Title* - An optional description of the gateway. If not specified, the IP address is used.
- *IP address* - The IP address of the gateway (required).
- *User ID/Password* - Some gateways require a user ID and password.

Delete

Removes the selected gateway. Note that no confirmation window is shown.

Timeout

The number of seconds that WapIDE will wait for a reply from the gateway.

Mode

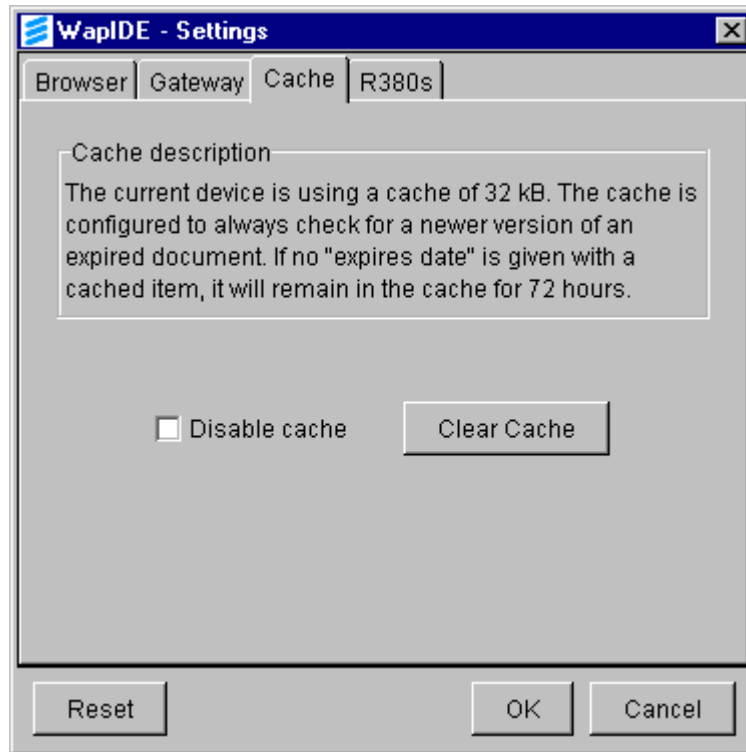
Use connectionless or connection-oriented sessions (not supported by the R320s device).

Security

Not used.

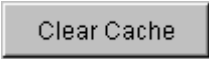
Cache settings

To improve performance, WapIDE keeps a non-persistent cache memory where the downloaded files (WML, WMLScript and images) are stored.



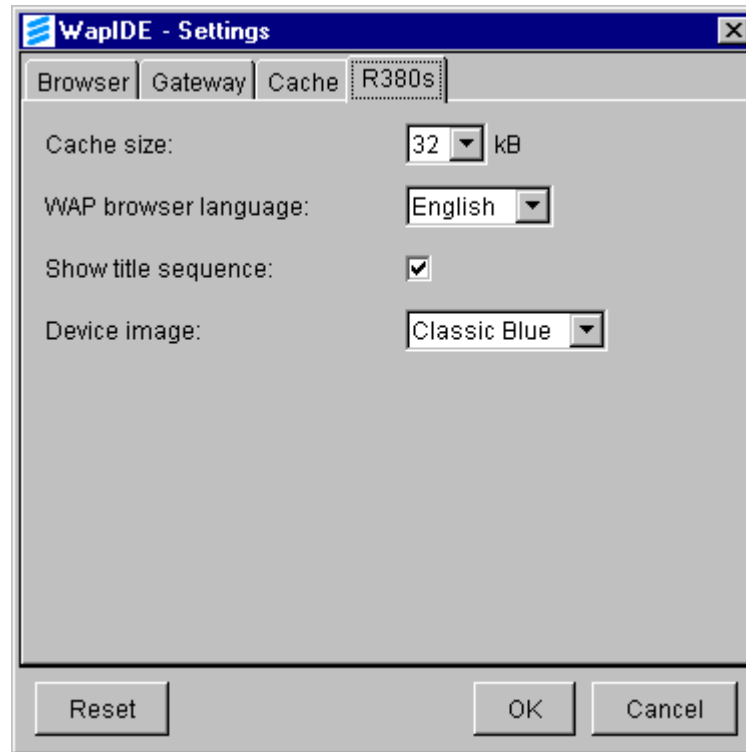
Cache description WapIDE uses the same cache size and options as the devices it simulates. This field describes the cache settings for the current device.

Disable cache Turns the cache off. While developing dynamic applications, it is recommended that you disable the cache to ensure that the decks are reloaded each time.

 Empties the cache memory.

Device-dependent settings

The options available in this tab depend on which device is currently active. The example below is for R380s.



Cache size

For the 380 device, you can select the size of the internal cache memory (0, 4, 8, 16 or 32 kB).

WAP browser language

The language can be set to English or (simplified) Chinese. Note that this only controls the language for the fixed texts in the device. The WapIDE windows are always in English.

See also *Testing applications in other character sets* on page 25.

Title bar sequence

For the 380 device, this indicates that the titles of the preceding cards in the application should be included in the window title (e.g. Card 1 > Card 2 > ...).

Device image

For each device, you can choose between different colors.

WAP browser font size

For the R520 device, you can specify the font size to use in the device display (small, medium, or large). This affects the number of lines that can be shown.

When the language is set to Chinese, the largest font size is always used.

Testing applications in other character sets

It is possible to test WML applications in other character sets, such as Chinese. Windows 2000 is recommended for developing international applications. Below, some aspects of using other character sets are described.

Standard texts

The standard texts in the device can be displayed in English or Chinese (see *Device-dependent settings* on page 24). To use the Chinese language you must have a *Unicode* font on your computer.

Windows 2000 has full support for East Asian languages built in. For other Windows versions, you must install a Unicode font and update the `Java font.properties` file.

WML presentation

WAP applications in any language can be presented in the browser. A Unicode font is required as described above.

East Asian text input

The Chinese models of the Ericsson phones allow entry of Chinese text from the device buttons. WapIDE does not support this.

Instead, you can use an *Input Method Editor* in Windows 2000 that allows you to enter Chinese text from the computer keyboard.

The default system locale must be set to the language you want to use. The Input Method Editor is started automatically when you enter text in input fields.



Using the WapIDE application designer

The WapIDE application designer is a WML editor with which you can design and test WAP applications. There is also a WMLScript editor for writing and compiling WMLScript code.

WML, WMLScript, and other files can be managed in projects.

The application designer is integrated with the WapIDE browser so you can easily test your applications on different devices.

Starting the application designer

Start the application designer by selecting

Programs ► Ericsson WapIDE 3.1 ► Application Designer

from the Windows Start menu.

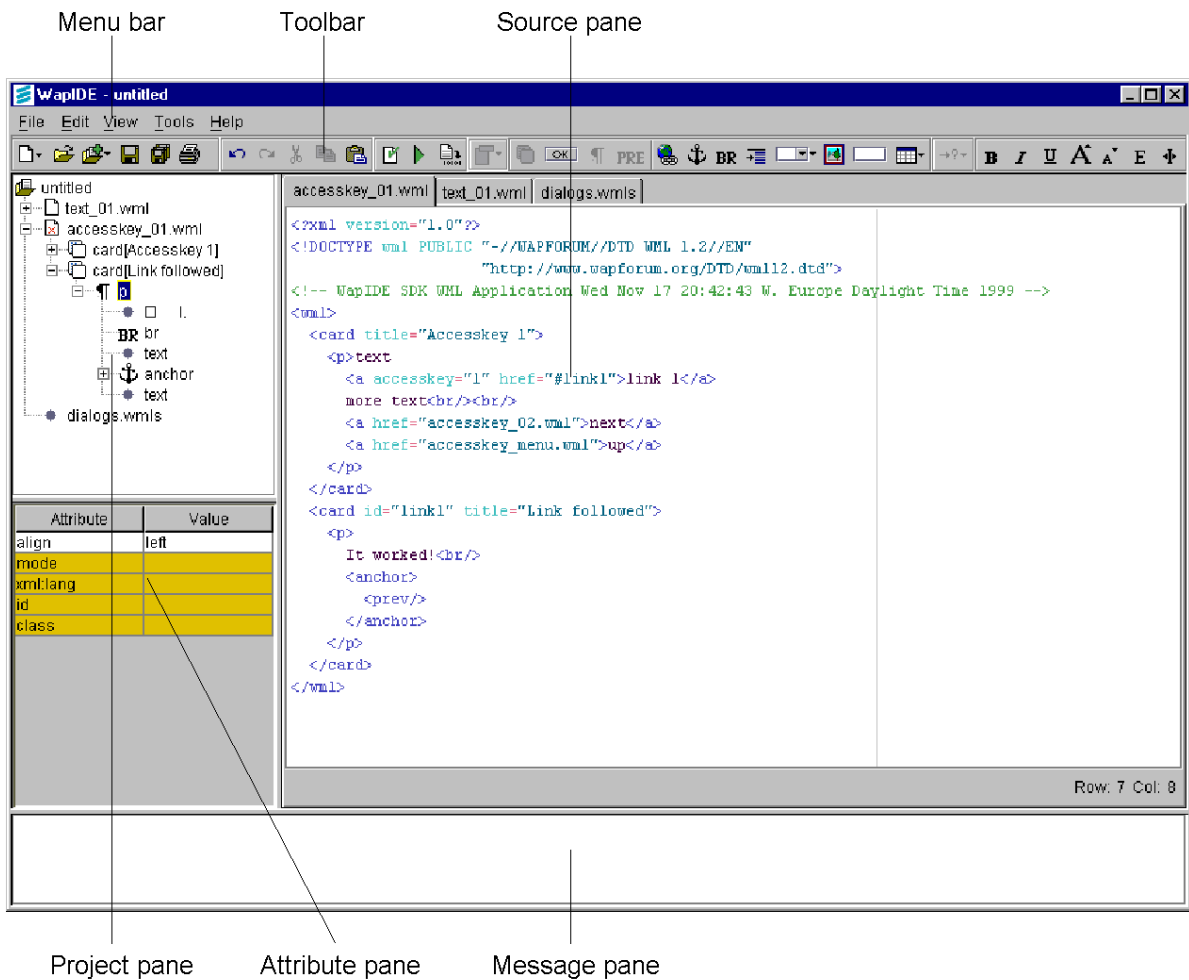
The application designer opens with the last project used. The first time it is started there is an empty project (“untitled”).

Application designer window

At the top of the application designer window is a menu bar and a toolbar. Below these are a number of different areas, or panes:

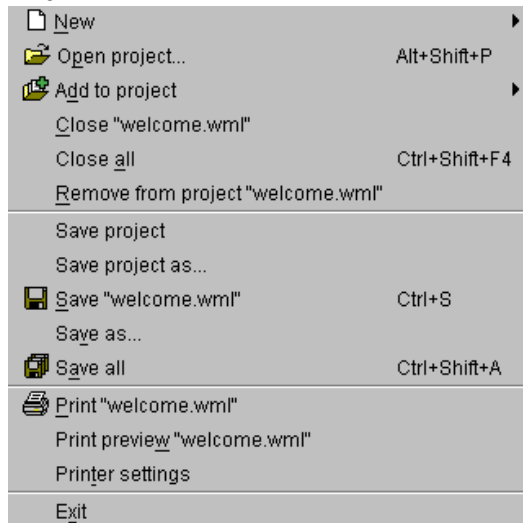
- The *project pane* shows a tree view of the current project, the files in the project, and the structure of the WML files.
- The *attribute pane* shows the attributes of the current project, file, or element in a WML file.
- The *source pane* shows the source code for the selected WML or WMLScript file. Each open file is shown in a separate tab.
- The *message pane* shows messages from compilation etc.

All panes can be resized and scrolled if necessary.

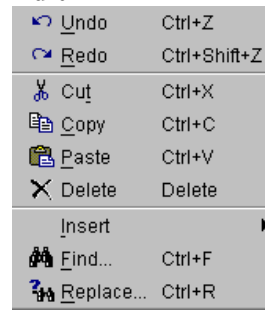


The following functions are available from the menu bar. Many of them are also available from the toolbar.

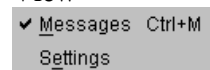
File



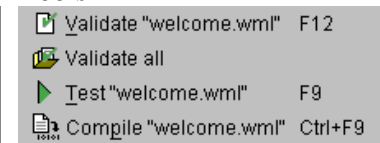
Edit



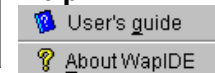
View



Tools



Help



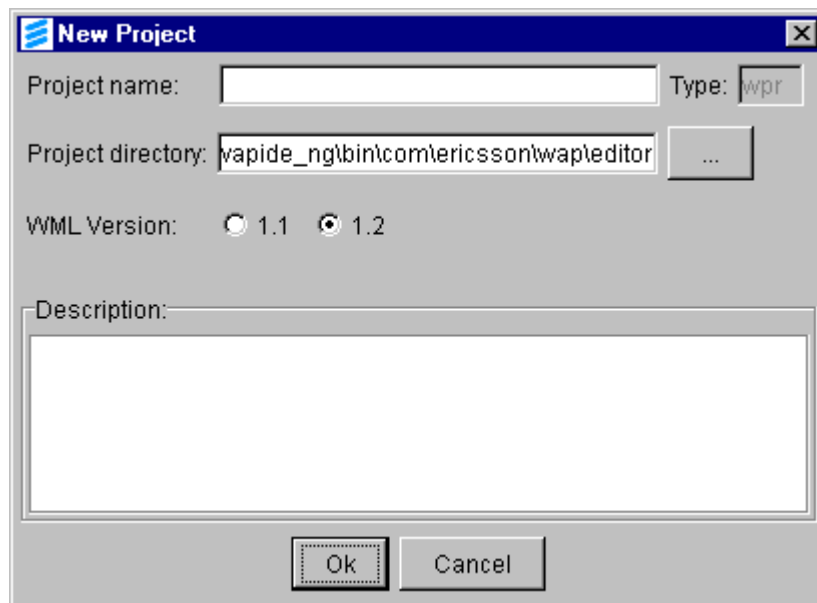
Working with projects

To simplify the administration of WAP applications, the source files are grouped into projects. Information about a project is saved in a project file (with the extension .wpr).

There is always one project active. If you don't want to work with files without storing them in a project, use the default project ("untitled") but never save it.

Creating a new project

To create a new project, select **File – New ► New project** from the menu bar. The following window is shown:




- Project name* The name of the project file.
- Project directory* The directory for the project file.
- WML version* The default WML version for files created in the project.
- Description* An optional description of the project.

The project properties are then shown in the attribute pane when the project is selected in the project pane.

Attribute	Value
comments	Default Project
currentDtd	1.2
projectLocation	!:\project\wapid...
projectName	untitled

Opening an existing project

To open an existing project, select **File - Open project...** from the menu bar or press the  icon and locate the project file.


Working with files

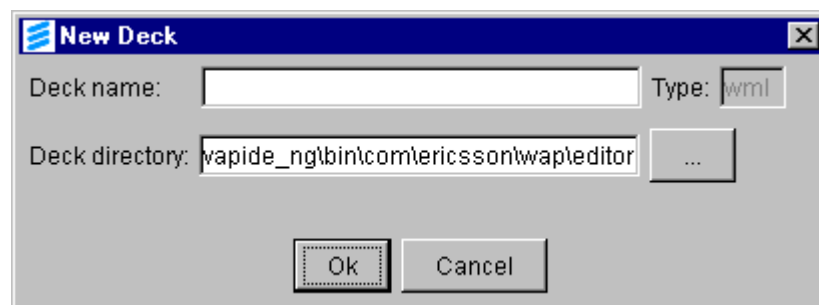
Multiple files may be open at the same time. Each file is shown in a separate tab in the source pane. Files are opened when you click on the file in the project pane.

When you hold the mouse over a file name in the project pane, the full path to the file is shown in pop-up window (“tool tip”).

In the source pane, normal text editing functions, such as cut & paste, undo/redo, and find & replace are available.

Creating new files


Select **File – New ► New WML deck** or **New WMLScript** or press the  icon to create a new file. The window below is shown.



A new file is automatically added to the current project.

When a new WML file is created, it contains the required header tags (?xml, !DOCTYPE, and wml) and a card with an empty paragraph.

Adding files to a project

To add a file to a project, select **File – Add to project ► Add WML deck** or **Add WMLScript** or press the  icon and locate the file.

When a WML file is added it is validated against the DTD (stored locally). If there are invalid tags, error messages are written but the parsing continues and the file is shown.

Other file types, including WMLScript (.wmls), pictures (.wbmp, .gif, and .jpeg) and different text files (.txt, .html, .jsp, and .asp) are shown as a single node in the project pane. The source code is shown if possible.

Removing files from a project

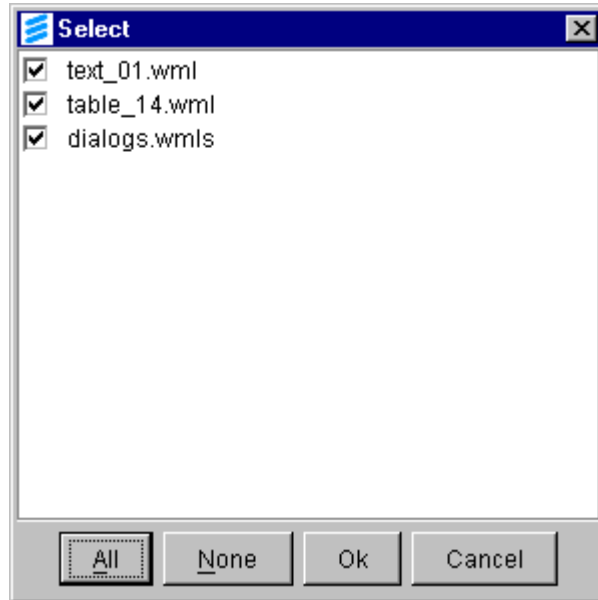
Select **File – Remove from project** to delete a file from the list in the project pane. The file itself is not changed.

Note that this works on the file selected in the project pane – not on the currently active tab like other menu options.

Saving and closing files

Modified files are marked with small red x in the file icon. You can save a selected file or all modified files.

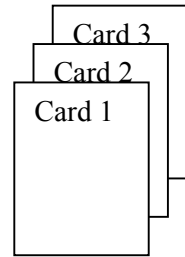
When you close files, or exit the application, WapIDE prompts you for the modified files.



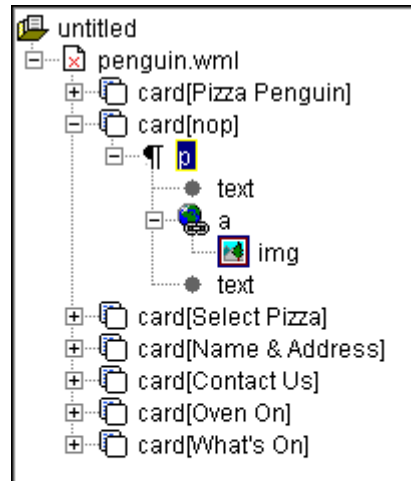
Editing WML decks

Structure of WML decks

A WML deck contains one or more cards and, optionally, a template. The template contains elements that are common to all cards. A typical example is a “Back” action that should be present in all cards.



The order of the cards does not matter except that the first card in the deck is the one that is displayed when the deck is loaded.



The structure of the WML deck is shown in a tree in the project pane. When you click on an element in the tree the source for that file is displayed and the attributes of the elements are shown in the attribute pane.

Attribute	Value
align	left
mode	
xml:lang	
id	
class	

In this example the p (paragraph) element has been selected and the attributes for the paragraph are shown. The only attribute specified (or defaulted) is align. The other attributes are shown in a different color to indicate that they have not been specified.

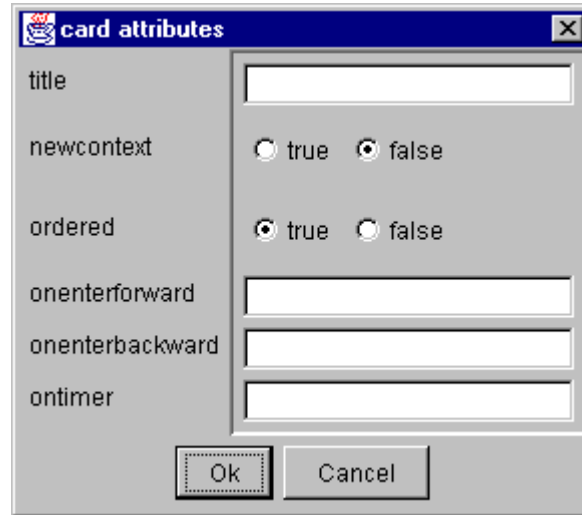
Inserting elements in a WML deck

From the toolbar (or by using **Edit – Insert ► ...** from the menu bar) you can insert new elements into a WML deck. The elements that are allowed at the current position are highlighted, the others are gray.

These toolbar icons are used to insert elements:

- | | | | |
|--|------------------------------|--|--|
| | template, head, meta, access | | br |
| | card | | fieldset |
| | do | | select, option, optgroup |
| | p | | image |
| | pre | | input |
| | a | | table, tr, td |
| | anchor | | timer, onevent, go, prev, noop, refresh, setvar, postfield |

When you insert an element, a pop-up menu with the element attributes is shown.



Three attributes (xml:lang, id, and class) are available on all elements but not often used. They are not shown in the attribute list.

Formatting text


The bold, italic, underline, big, small, emphasis, and strong icons on the toolbar are used to format selected text parts.



Select the text to be formatted and press the icon (or select **Edit – Insert ► Markup ► ...** from the menu bar).

Validating a WML deck

When you open a WML file the WML structure in the project pane and the element attributes are created. As you modify the WML source code there is no automatic update of the project and attribute panes.

To verify the syntax of a WML file, and update the project and attribute panes, select **Tools – Validate** or press the  icon. If there are syntax errors, messages are shown in the message pane.

```
welcome.wml, ** FatalError **, The end-tag for element type "p" must end with a '>' delimiter. Line: 14 Column: 1
```

You can click on a message to highlight the line with the syntax error.

You can also validate all WML files in the project by selecting **Tools – Validate all**.

Encoding a WML deck


WML files can be encoded (i.e. compiled) to verify the syntax and to create a compiled (.wmlc) file.

Select **Tools – Compile** or press the  icon to compile a file. If it is modified, the file is automatically saved before compilation.

Compilation information and error messages are shown in the message pane. The compiled file is stored in the same directory as the source file.

Editing WMLScript files

The WMLScript editor is a standard text editor.


WMLScripts can be compiled to verify that the syntax is correct and to create a compiled (.wmlsc) file. Select **Tools – Compile** or press the  icon to compile a file. Compilation messages are shown in the message pane.

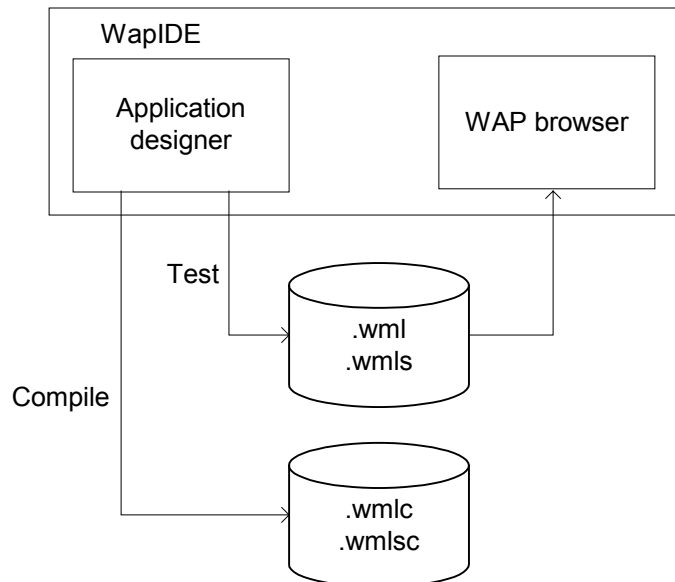
```
[Compiler] Read 166 bytes into buffer  
[Compiler] Compiling WMLScript file: E://calc.wmls  
[Compiler] The result is stored in E://calc.wmlsc  
[Compiler] The file size is 51 bytes
```

The compiled file is stored in the same directory as the source file.

As with WML files, it is not necessary to compile WMLScripts to test them in the WAP browser but it is the only way to test the syntax since there is no validate function for WMLScript.

Testing applications

For WML decks, select **Tools – Test** or press the  icon to display the current deck in the WAP browser window. If the file has been modified, it is saved before the browser is invoked.



Note: It is **not** necessary to compile the files to test them in the WAP browser. When you test an application the browser automatically compiles WML and WMLScript to temporary files.

Creating applications in other character sets

To create applications in other character sets than ASCII, you should specify the xml encoding parameter (on the first line in each WML deck), for example:

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

The WML editor checks the encoding parameter when it reads and writes WML files.

Note: If no encoding is specified, the system default encoding is used. This means that the encoding can change if you move your files to another machine.

The character sets supported in the WML editor depends on the support in Java and the operating system. When you compile or test your applications in the WAP browser, the following character sets are supported:

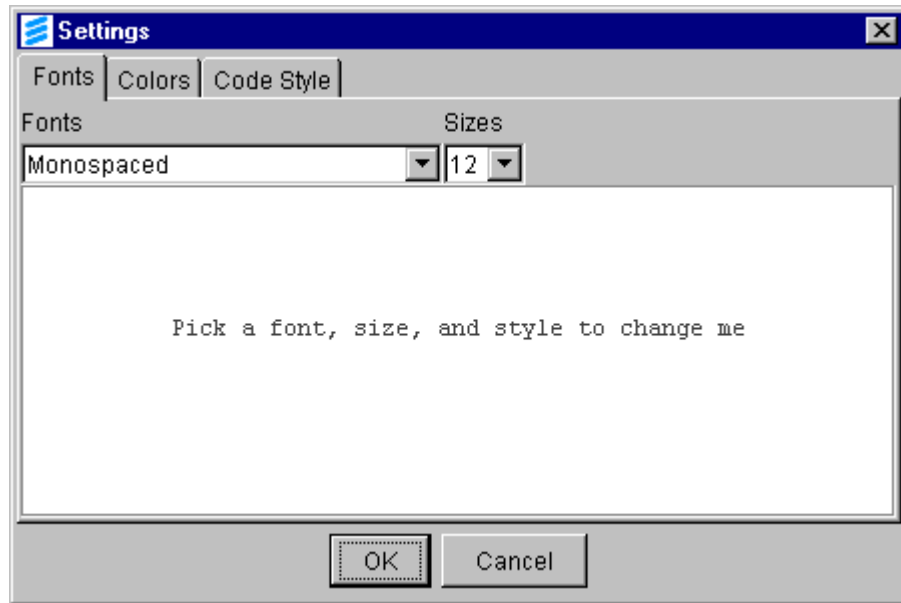
Type	Character set	Aliases
Unicode	iso-10646-ucs-2	
	iso-10646-ucs-4	
	utf-8	
	utf-16	
	us-ascii	
Western European	iso-8859-1	cp819, csisolatin1, ibm819, iso_8859-1, iso8859-1, iso-ir-100, 11, latin1
	Central European	iso-8859-2
	windows-1250	
Baltic	windows-1257	
Arabic	iso-8859-6	arabic, csisolatinarabic, ecma-114, iso_8859-6, iso-ir-127
Greek	iso-8859-7	csisolatingreek, ecma-118, elot_928, greek, greek8, iso_8859-7, iso-ir-126
	windows-1253	
Turkish	iso-8859-9	csisolatin9, iso_8859-9, latin9
Japanese	shift_jis	ms_kanji, csshiftjis
	euc-jp	extended_unix_code_packed_format_for_japanese, cseucpkdfmtjapanese
	iso-2022-jp	csiso2022jp
Korean	ks_c_5601-1987	
	euc-kr	
Chinese	gb2312	
	big5	
Thai	windows-874	cp874

All character set names are case insensitive.

Setting application designer preferences

These settings control the text presentation in the application designer source pane.

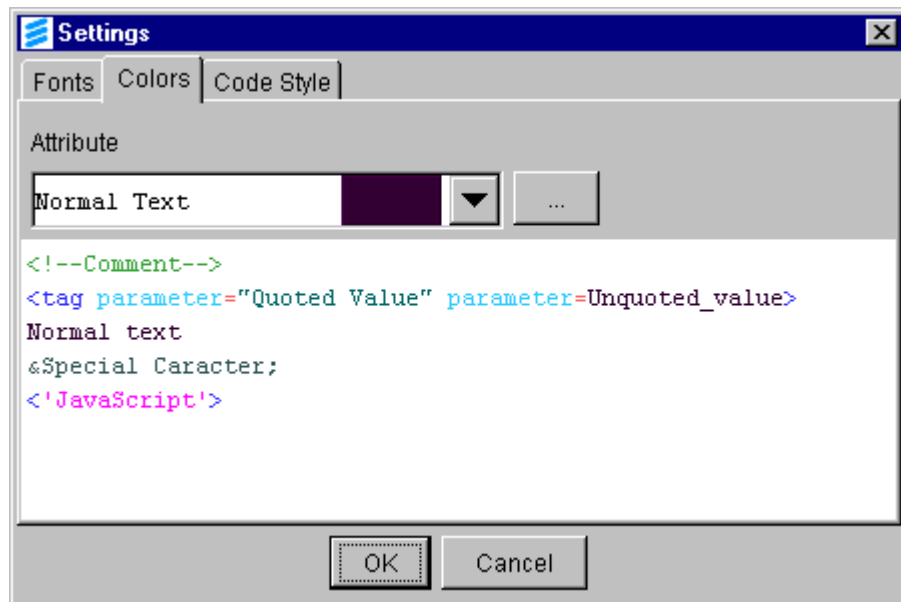
Fonts



Font Select the font to use from the list.

Size Select the font size from the list.

Colors

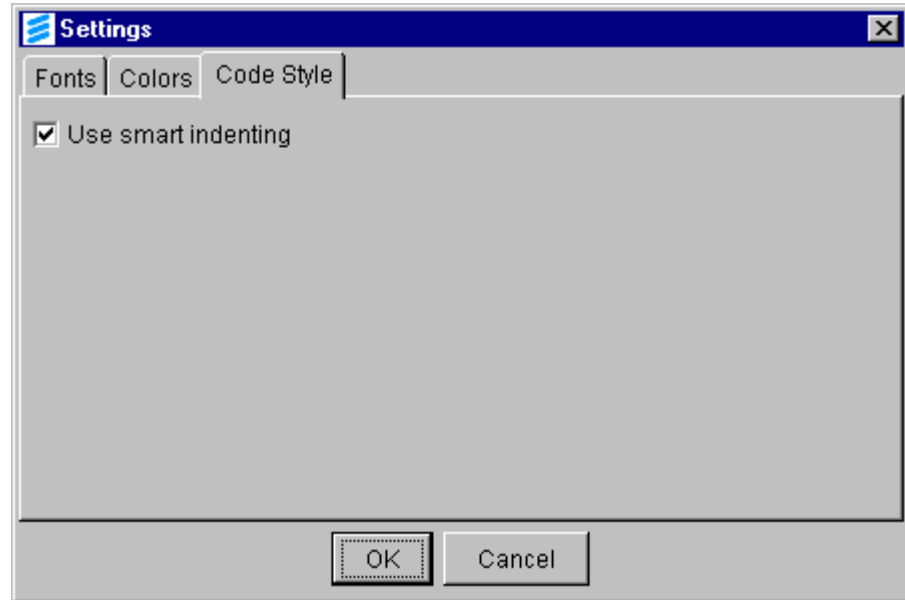


Attribute Select the attribute type to change the color for from the list.



Press this button to open a color chooser for the selected attribute.

Code style

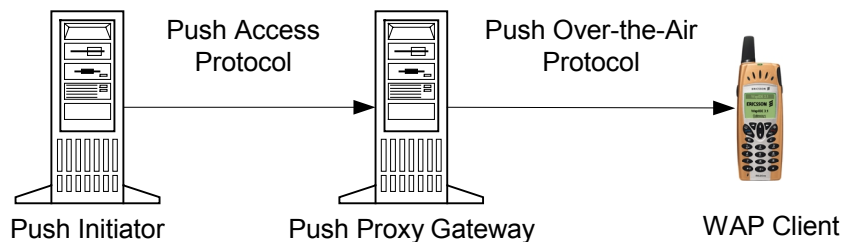


Use smart indenting

When this box is checked, new lines are indented at the same position as the previous line.

Working with push messages

Push is a new WAP feature. In the traditional “pull” case the transactions are always initiated from the client. Push technology allows a *push initiator*, somewhere on the Internet, to send content to a WAP client via a *push proxy gateway* (PPG).



The push initiator sends content to the PPG using the *push access protocol* (PAP). The PPG delivers the push message to the WAP client using the *push over-the-air protocol* (OTA).

PAP uses XML documents that are sent over HTTP. The following operations are defined:

- Push submission (initiator to PPG)
- Result notification (PPG to initiator, not supported by WapIDE)
- Push cancellation (initiator to PPG)
- Status query (initiator to PPG)
- Client capabilities query (initiator to PPG).

PAP allows any content type to be pushed to the WAP client. However, the client may not be able to receive them. The WAP standard defines three push content types:

- Service Indication (SI) – send a notification to the WAP client.
- Service Loading (SL) – cause the WAP client to load and execute a service.
- Cache Operations (CO) – invalidate content objects in the WAP client cache.

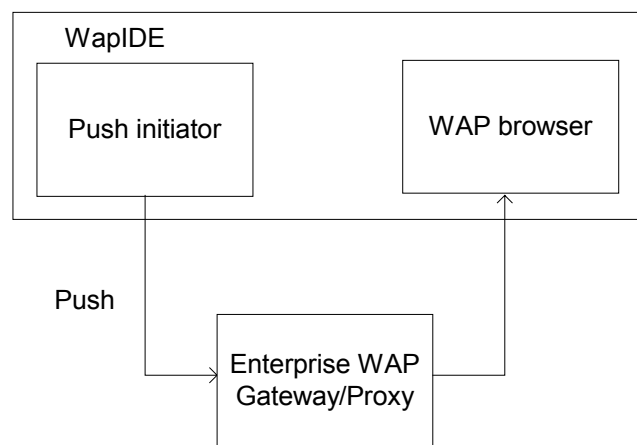
Different security levels can be used for authenticating the push initiator, such as TLS/SSL, signed or encrypted content, HTTP authentication, or no authentication (in a trusted network).

For a complete description of the WAP push framework, refer to the specifications on www.wapforum.org.

WapIDE implementation of push

WapIDE supports push in the following way:

- WapIDE provides a push initiator where push messages can be created and sent to a push proxy gateway using PAP.
- Normally, the PPG sends the messages to a mobile phone. The Ericsson Enterprise WAP Gateway/Proxy 2.0 (or WAP Gateway/Proxy 3.0) can be configured to instead send the messages to the WapIDE browser (using an internal protocol). Refer to *Installing a local gateway* on page 55 for configuration instructions.
- The WapIDE browser can receive and display push messages in the same way as the real devices. Push is currently supported by the R520m device only.



You can also use the WapIDE push initiator to push messages to a real terminal, or use other push initiators with the WapIDE browser.

Note: When the gateway is configured to send push messages to WapIDE, all messages will be sent to one WapIDE browser regardless of the address specified in the push message. To push messages to another WapIDE browser, or to a real terminal, the configuration must be changed.

The WapIDE push initiator supports two security levels:

- No security
- HTTP basic authentication.

Using the push initiator

Start the push initiator by selecting

Programs ► Ericsson WapIDE 3.1 ► Push Initiator

from the Windows Start menu.

The screenshot shows the 'Push Initiator - WapIDE' application window. The title bar includes the application name and standard window controls. The menu bar contains 'File', 'View', and 'Help'. The main window is titled 'WapIDE Push Message Initiator' and is divided into three main sections: 'Push Operation', 'Content Definition', and 'PPG Communication'.

Push Operation

This section contains four tabs: 'Push Submission', 'Cancellation', 'Status Query', and 'Client Capabilities Query'. The 'Push Submission' tab is active. It features a 'Push ID' field with the value 'WapIDE_3.0_PI/1'. Below it is an 'Address' section with a text input field (currently empty) and an 'Add' button. A list box below the input field contains the number '+11111111', with a 'Remove' button to its right. An 'Advanced..' button is located at the bottom right of this section.

Content Definition

This section contains five tabs: 'Push Message', 'WAP SI', 'WAP SL', 'WAP CO', and 'User defined'. The 'WAP SI' tab is active. It features a 'Push Message Contents' section with a table:

Part	Content Type	Content Description	Content Size
1	text/vnd.wap.si	WAP Service Indication	232

Below the table is a 'Delete Part' button and an 'Advanced...' button.

PPG Communication

This section contains a text area with the following XML code:

```
<?xml version="1.0" ?>
<!DOCTYPE pap PUBLIC "-//WAPFORUM//DTD PAP 1.0//EN" "http://www.wapforum
<pap>
<push-message push-id="WapIDE_3.0_PI/0">
  <address address-value="wappush="+11111111/type=plmn@eip.ericsson.se"/>
</push-message>
```

At the bottom of the window is a 'Send Message' button.

The WapIDE push initiator allows you to create push messages and send them to a push proxy gateway. It also receives and displays responses from the gateway.

The push initiator window has three main areas:

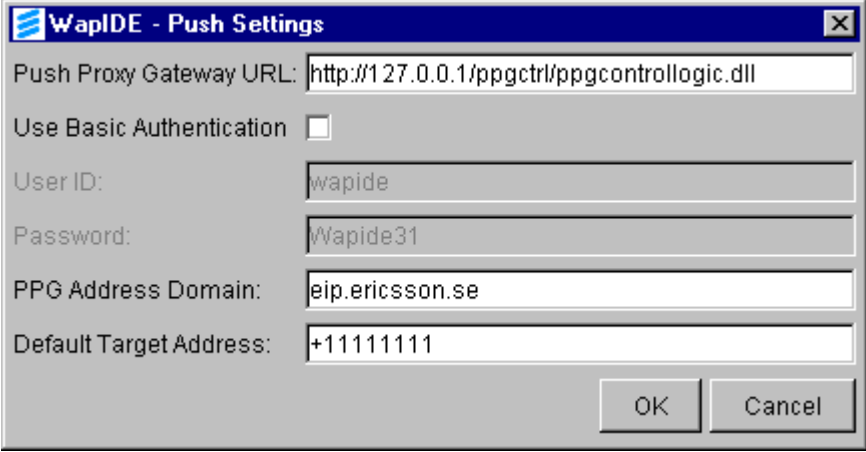
- **Push Operation** – defines the type of push operation and control information that the PPG uses to send the message to the correct client.
- **Content Definition** – the push message content intended for the WAP client (only present for push submission operations). The push message can consist of one or more parts.
- **PPG Communication** – shows the messages sent to and responses received from the PPG in XML format.

The following steps are used to send a push message to the WapIDE browser (or another client):

1. Open the push settings window and enter the correct PPG information. See *Push settings* on page 42 for more information.
2. Select the Push Submission tab and enter the PAP header information for the PPG. See *Push operation* on page 42.
3. Define the content to be sent to the WAP client. It may consist of one or more parts. See *Content definition* on page 45.
4. Send the message by pressing the **Send Message** button or by using **File – Send** from the menu bar. You can also look at the PAP message source before it is sent by using **File – Preview**.
5. Check the response messages from the gateway. See *PPG communication* on page 49.
6. Check the result in the browser. See *Receiving push messages* on page 49.

Push settings

Select **View – Settings** from the menu bar to open the Push Settings window. These settings must match the configuration of the gateway (see *Configuring the gateway for push* on page 55).



<i>Push proxy gateway URL</i>	The URL for posting push messages. For Enterprise WAP Gateway/Proxy, it should be specified as <a href="http://<hostname>/ppgctrl/ppgcontrollogic.dll">http://<hostname>/ppgctrl/ppgcontrollogic.dll
<i>Use basic authentication</i>	Indicates that HTTP basic authentication should be used when posting to the PPG. Otherwise, no security is used.
<i>User ID</i> <i>Password</i>	If basic authentication is used, the user ID and password must be specified.
<i>PPG address domain</i>	The Internet host name of the PPG. Ignored by Enterprise WAP Gateway/Proxy.
<i>Default target address</i>	To simplify testing, this address is always part of the address list initially.

Push operation

The following operations are supported by the WapIDE push initiator:

- **Push submission** – send a push message to the PPG.
- **Cancellation** – cancel a previously submitted push message.
- **Status query** – request current status of previously submitted push message.
- **Client capabilities query** – query the PPG for the capabilities of a specific device.

Note that Ericsson Enterprise WAP Gateway/Proxy does not support cancellation and client capabilities query.

For **push submission**, **cancellation**, and **status query**, the following can be specified:

The screenshot shows a dialog box titled "Push Operation" with four tabs: "Push Submission", "Cancellation", "Status Query", and "Client Capabilities Query". The "Push Submission" tab is active. It contains a "Push ID:" field with the value "WapIDE_3.0_PI/1". Below it is an "Address:" field with the value "+11111111". To the right of the address field are two buttons: "Add" and "Remove". At the bottom right of the dialog is an "Advanced.." button.

Push ID A unique ID of the push message. It can be used to cancel or check status on the message. WapIDE generates this ID automatically but it can be changed by the user.

Address One or more receivers of the message. This should contain the subscriber MSISDN number.

For cancellation and status query, this field is optional. If omitted, the request applies to all messages with the specified push ID.

For **client capabilities query**, these fields can be specified:

The screenshot shows the same "Push Operation" dialog box, but with the "Client Capabilities Query" tab selected. The "ccq-message" label is visible at the top of the main content area. The "Query ID:" field contains "WapIDE_3.0_PI/0". The "Application ID:" field is empty. The "Address:" field contains "+11111111".

Query ID An optional identification of the query that is returned in the response.

Application ID An optional ID of the application in the client that will be the target for subsequent push messages.

Address The target device address (subscriber MSISDN number).

Push submission - advanced settings

The push submission operation has additional options that can be accessed using the **Advanced...** button. Many of these are not supported by Ericsson Enterprise WAP Gateway/Proxy (EWGP).

Push Message - Advanced Settings

Push Message Fields

Deliver Before [YYYY-MM-DD]: 2001 - 4 - 7 [hh:mm:ss]: 13 : 38 : 25 Include

Deliver After [YYYY-MM-DD]: 2001 - 4 - 6 [hh:mm:ss]: 13 : 38 : 25 Include

Source Reference:

ppg-notify-requested-to:

progress-notes-requested

Quality of Service Fields

Priority

Delivery Method

Network: Required

Bearer: Required

OK

- Deliver before* The date and time by which the content must be delivered to the WAP client. The default is one day from the current time. Check *Include* to add this field. Not supported by EWGP.
- Deliver after* The date and time after which the content should be delivered to the WAP client. The default is the current date and time. Check *Include* to add this field. Not supported by EWGP.
- Source reference* A textual name of the content provider. This is useful to a PPG operator in identifying the originator of the message. Ignored by EWGP.
- Ppg-notify-requested-to* The address (e.g. URL) that the PPG should use for notification of results related to this message. Note that the WapIDE push initiator can not receive these notifications.
- Progress-notes-requested* Informs the PPG as to whether or not the push initiator wants to receive progress notes. Ignored by EWGP.
- Priority* The delivery priority of the message (low, medium, or high). Ignored by EWGP.
- Delivery method* The desired OTA delivery method. For EWGP, “notspecified”, “unconfirmed”, and “preferconfirmed” all result in unconfirmed push. “Confirmed” is not supported by EWGP.
- Network* The network desired for use when delivering the message. If *Required* is checked, the specified network must be used. EWGP only supports “GSM”.
- Bearer* The bearer desired for use when delivering the message. If *Required* is checked, the specified bearer must be used. EWGP only supports “SMS”.

Content definition

The push message contents to be sent to the WAP client typically consists of a single WAP SI or SL message. It may also be a multi-part message, with e.g. a WAP CO message followed by a WAP SL message (note that multi-part messages are not supported by Enterprise WAP Gateway/Proxy).

Part	Content Type	Content Description	Content Size
1	text/vnd.wap.si	WAP Service Indication	232

The *Push Message* tab in the content definition area contains a list of all the parts. The other tabs are used to create parts of the message.

Use this button in the *WAP SI*, *WAP SL*, *WAP CO*, and *User defined* tabs to create message parts.

Use this button in the *Push Message* tab to delete parts.

Advanced content settings

Additional HTTP header settings, that apply to all parts of the content, can be accessed using the **Advanced...** button.

Advanced Content Settings

Additional WAP HTTP Headers

X-Wap-Application-Id:

X-Wap-Content-URI:

X-Wap-Initiator-URI:

OK

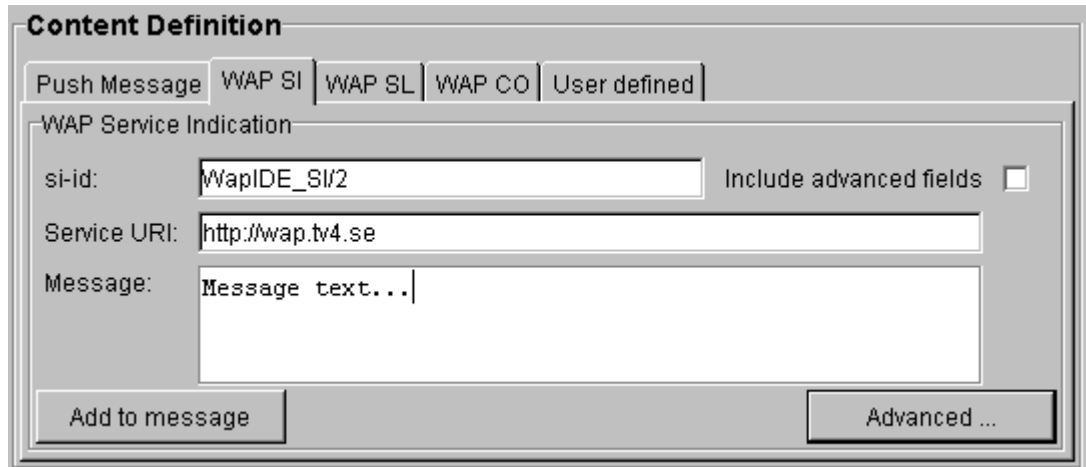
X-Wap-Application-Id
X-Wap-Content-URI
X-Wap-Initiator-URI

See the *WAP Push Message* specification on www.wapforum.org for a description of these HTTP headers.

WAP SI

The Service Indication (SI) content type provides the ability to send notifications to end-users in an asynchronous manner. Such notifications may, for example, be about new e-mails, changes in stock price, news headlines, etc.

In its most basic form, an SI contains a short message and a URI indicating a service.

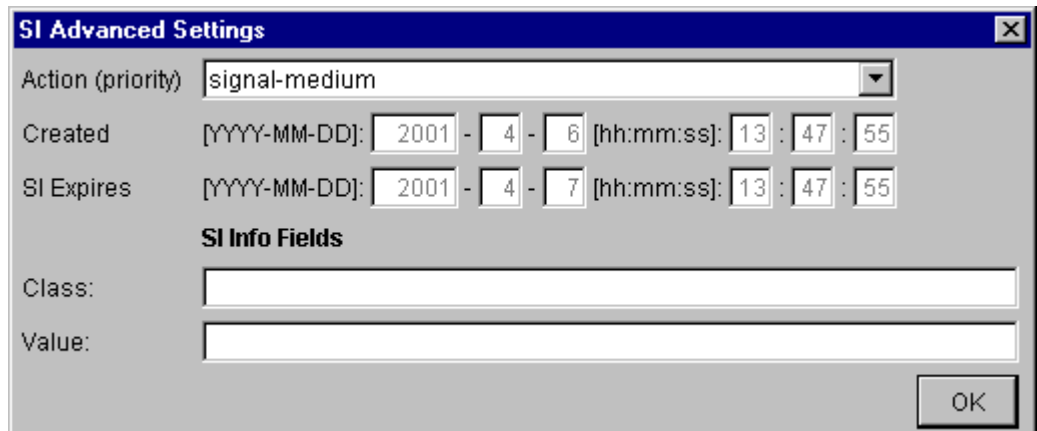


The image shows a 'Content Definition' dialog box with several tabs: 'Push Message', 'WAP SI', 'WAP SL', 'WAP CO', and 'User defined'. The 'WAP SI' tab is selected. The dialog is titled 'WAP Service Indication'. It contains three input fields: 'si-id:' with the value 'WapIDE_SI/2', 'Service URI:' with the value 'http://wap.tv4.se', and 'Message:' with the value 'Message text...'. There is a checkbox labeled 'Include advanced fields' which is currently unchecked. At the bottom of the dialog, there are two buttons: 'Add to message' and 'Advanced ...'.

- SI ID* May provide the SI with an identity in order to make it possible to distinguish between different SIs. If not specified, it is considered to be the same as the value of *Service URI*.
- Service URI* This attribute specifies the URI that is used to access the service. If no service URI is specified, the SI corresponds to a notification (no service can be initiated).
- Message* The text presented to the end-user upon reception.
- Include advanced fields* Add the fields under advanced settings to the message.

SI advanced settings

Using the **Advanced...** button, the following fields can be specified:



The image shows a 'SI Advanced Settings' dialog box. It has a title bar with a close button. The dialog contains several fields: 'Action (priority)' is a dropdown menu set to 'signal-medium'; 'Created' is a date and time field showing '2001 - 4 - 6 [hh:mm:ss]: 13 : 47 : 55'; 'SI Expires' is a date and time field showing '2001 - 4 - 7 [hh:mm:ss]: 13 : 47 : 55'. Below these is a section titled 'SI Info Fields' with two empty text input fields labeled 'Class:' and 'Value:'. An 'OK' button is located at the bottom right of the dialog.

<i>Action (priority)</i>	<p>The action to be taken when the SI is received:</p> <ul style="list-style-type: none"> • Signal low/medium/high – Present the message to the user. The exact behavior for the different actions depends on the client. • Signal none – The SI will not be presented to the end-user. However, the client may use the information carried in the SI info fields to perform certain tasks. • Delete – The received SI and any other SI with identical <i>SI ID</i> will be deleted (<i>SI ID</i> must be specified).
<i>Created</i>	The date and time of creation or last modification of the content indicated by <i>Service URI</i> .
<i>SI expires</i>	The date and time when the SI expires and thereby is automatically deleted or marked as “expired”. If this attribute is not specified, the SI never expires.
<i>Class Value</i>	The SI info fields provide a means to specify additional information to the WAP client. Refer to <i>WAP Service Indication</i> on www.wapforum.org for more information.

WAP SL

The Service Loading (SL) content type provides the ability to cause a user agent on a mobile client to load and execute a service that, for example, can be in the form of a WML deck.

The screenshot shows a 'Content Definition' dialog box with the following elements:

- Tabbed interface: Push Message, WAP SI, **WAP SL**, WAP CO, User defined
- Title: WAP Service Loading
- Service URI:
- Action: (dropdown menu)
- Button:

<i>Service URI</i>	The URI that is used to access the service.
<i>Action</i>	<p>The action to be taken when the SL is received.</p> <ul style="list-style-type: none"> • Execute low/high – The service identified by the URI is loaded (either from an origin server or from the client’s cache) and executed. The exact behavior for the different actions depends on the client. • Cache - The service is loaded in the same way as above but placed in the cache instead of being displayed in the browser.

WAP CO

The Cache Operations (CO) content type provides a means to invalidate content objects in the user agent cache. The invalidated content objects must be reloaded from the origin server the next time they are needed.

There are two operations: *invalidate object* and *invalidate service*,

- **invalidate object** - invalidate the object uniquely identified by the given URI.
- **invalidate service** - invalidate all the objects that share the same URI prefix.

The screenshot shows a dialog box titled "Content Definition" with several tabs: "Push Message", "WAP SI", "WAP SL", "WAP CO", and "User defined". The "WAP CO" tab is selected. Below the tabs, the text "WAP Cache Operations" is displayed. There are two input fields: "Invalidate Object:" with the value "http://" and "Invalidate Service:" with the value "http://". At the bottom right, there is a button labeled "Add to message".

Invalidate object The URI of the cached object to invalidate.


Invalidate service The URI prefix of the cached objects to invalidate.

User defined

Other content types for specific applications may be added under the *User defined* tab.

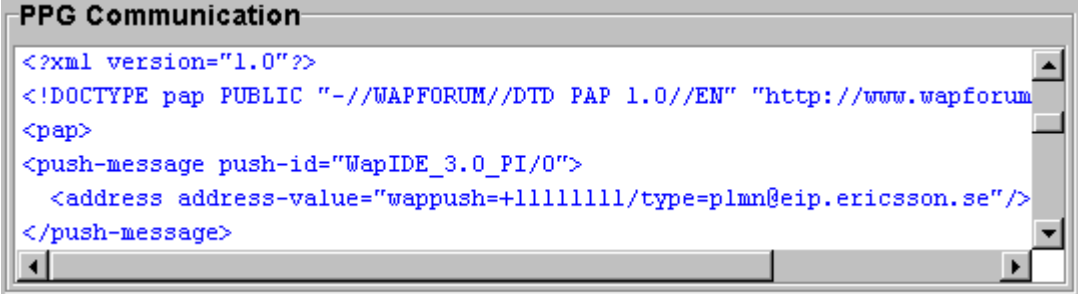
The screenshot shows a dialog box titled "Content Definition" with several tabs: "Push Message", "WAP SI", "WAP SL", "WAP CO", and "User defined". The "User defined" tab is selected. Below the tabs, the text "User-defined Content Type" is displayed. There is a "Content type:" dropdown menu with the value "text/vnd.wap.wml" and a "Content file:" text input field with a browse button ("...") to its right. At the bottom right, there is a button labeled "Add to message".

Content type Select a content type from the list or type in another content type.

Content file Enter a file name or use the  button to locate the file.

PPG communication

The PPG communication field shows the messages sent to and responses received from the push proxy gateway in XML format.



```
PPG Communication
<?xml version="1.0"?>
<!DOCTYPE pap PUBLIC "-//WAPFORUM//DTD PAP 1.0//EN" "http://www.wapforum
<pap>
<push-message push-id="WapIDE_3.0_PI/0">
  <address address-value="wappush="+11111111/type=plmn@eip.ericsson.se"/>
</push-message>
```

The following colors are used for the text in the PPG communication field:

- Blue Messages sent to the PPG.
- Green Messages received from the PPG.
- Red Error messages (such as HTTP errors).

Receiving push messages

The WapIDE browser can receive and display push messages in the same way as the real devices. An exception is the push inbox function that allows you to look at old push messages. The WapIDE browser does not support this function.

Push is currently supported by the R520m device only.

Under the *Push* tab in the browser trace window (see *Using the browser trace* on page 20), you can see details about received push messages.

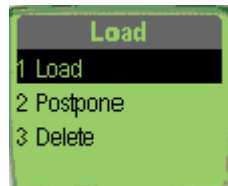
Service indication

When the browser receives a WAP SI message, a window like this is shown:



The text is the user message specified on the SI message (the title is the first part of the text).

When "Proceed?" is selected, this window is shown:



Load loads the URI specified in the SI message. Delete removes the message. Postpone is not supported since WapIDE does not store old push messages.

Other message types

The other content types are handled in the following way:

- **Service loading** – the specified URI is loaded and displayed in the browser immediately.
- **Cache operations** – these operations are not visible for the user.
- **User-defined** – not supported by the WapIDE browser.

Pushing messages to a real terminal

You can also use the WapIDE push initiator to push messages to a real terminal (provided that your gateway is connected to a network that supports push).

In this case, you should not reroute push messages from the gateway to WapIDE (as described in *Define receiver IP address* on page 57). It is also important that a real MSISDN is defined (see *Define subscriber* on page 57).

Writing applications that generate push messages

You can use WapIDE as a help in creating push applications.

1. Use the WapIDE push initiator to create the push messages you want.
2. Copy the XML messages from the PPG communication window and use them as a model.
3. Write the application that creates the push messages and sends them to the gateway using HTTP post. Note that you need to use HTTP basic authentication (or a higher security) if you don't deactivate that in the gateway.
4. Test the application by pushing messages to the WapIDE browser.
5. When the application is working, deploy it in a real environment.

Setting up a WAP application environment

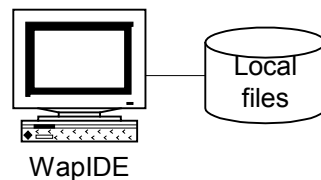
Overview

The WapIDE browser, like a mobile phone, accesses content from a web server via a WAP gateway. WapIDE can also read local files without a gateway. In the following, three scenarios are described:

1. Accessing local WML files. This is an easy way to test static applications.
2. Using a gateway on the Internet. This method can be used to test applications that are available on the Internet.
3. Using a local gateway. If you want to test local applications on a web server, or if a firewall prevents you from using a gateway on the Internet, you have to install a local test gateway.

Accessing local WML files

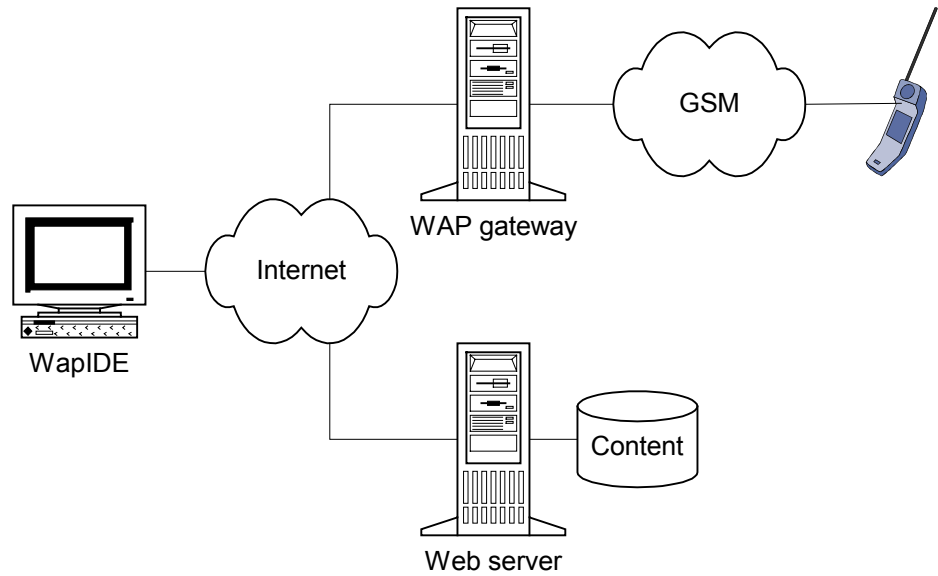
Static, local WML applications can be tested in the WapIDE browser without using a web server or a gateway. Place your WML files on a local disk and access them with <file:///path/file.wml>.



Local files can use relative references to other files so you can develop an application and then move the directory structure to a web server.

Using a gateway on the Internet

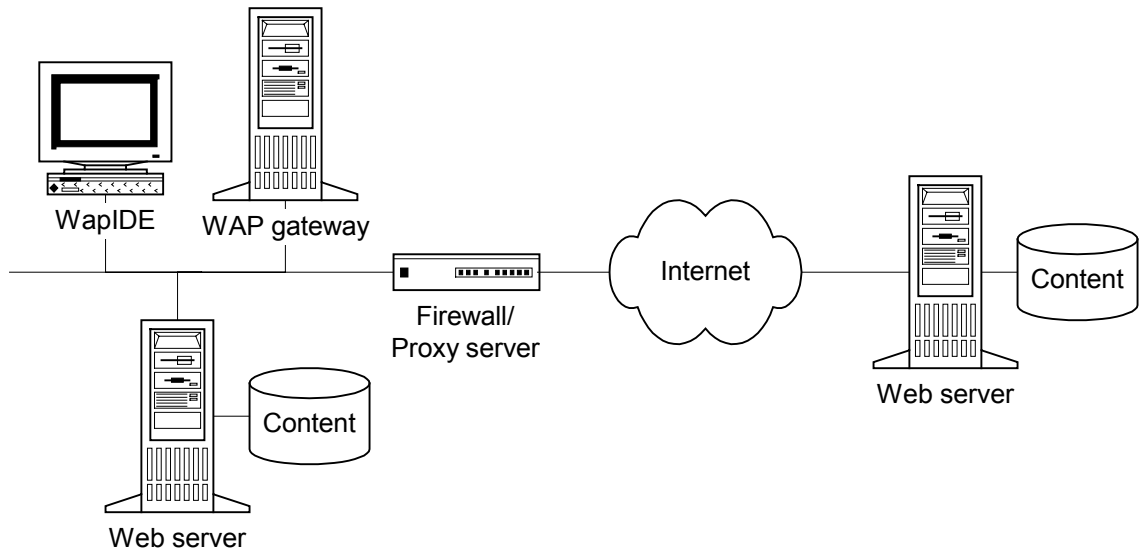
To access web servers on the Internet, you can use a WAP gateway on the Internet. Ericsson provides a WAP gateway for external test use. Refer to the Ericsson Developers' Zone for more details. Note that the IP communication between WapIDE and the WAP gateway uses UDP ports 9200 and 9201. If you have any firewalls between WapIDE and the WAP gateway, these ports must be opened in the firewall.



Using a local gateway

If you want to test your own local web server applications that are not available on the Internet, or if you are placed behind a firewall that prevents you from communicating with external gateways, you need to install a local test gateway.

The local gateway can also be used to access web servers on the Internet. If your local network is protected by a firewall you must then specify an HTTP proxy server to use.



A demo version of Ericsson's Enterprise WAP Gateway/Proxy 2.0 can be downloaded from the Ericsson Developers' Zone. Refer to *Installing a local gateway* on page 55 for more information.

Configuring a web server

When you place your WAP application on a web server you have to configure it to support the right MIME types. Refer to the documentation of your web server for information on how to do this.

Content type	MIME type	File extension
WML source	text/vnd.wap.wml	wml
WMLScript source	text/vnd.wap.wmlscript	wmls
Compiled WML	application/vnd.wap.wmlc	wmlc
Compiled WMLScript	application/vnd.wap.wmlscriptc	wmlsc
Wireless bitmap	image/vnd.wap.wbmp	wbmp

Creating dynamic WML applications

In dynamic applications, the WML decks are generated from input given by the user, contents of a database, etc. This is typically done with JavaServer Pages (JSP), Java servlets, Active Server Pages (ASP), or CGI scripts. Testing these applications requires a web server with support for the chosen method.

It is important that the response content type is set correctly in the generated WML. Here is a simple JSP example:

```
<% response.setContentType("text/vnd.wap.wml"); %>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
    "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="card1">
  <p>JSP test...</p>
</card>
</wml>
```

In ASP, the following is used instead:

```
<% Response.ContentType = "text/vnd.wap.wml" %>
```

Managing sessions

Dynamic applications often require that the web server keeps session information, which is usually handled with *cookies*. Most WAP devices do not support cookies. Some WAP gateways (e.g. Ericsson's WAP Gateway/Proxy) support cookies and store them on the client's behalf.

This means that you cannot assume that cookies are supported in the WAP environment and that you have to use other methods for session management, such as including a session ID in the decks ("URL rewriting").

Creating WMLScript applications

WMLScript is used together with WML to create WAP applications. It places some procedural logic at the client side and thus reduces the need for communication with the server. Typical usage is to validate user input and access function libraries stored in the client.

To use WMLScript you must first create your own WMLScript and then call it from a WML file, as in the following example.

```
<wml>
  <card>
    <p>Enter amount:<input type="text" format="*N" name="N" />
    <br/>
    Total = $Sum
  </p>
  <do type="accept">
    <go href="calc.wmls#calcInterest($N,12)" />
  </do>
</card>
</wml>
```

The file `calc.wmls` contains the WMLScript code:

```
extern function calcInterest(N,r) {
  var Total;
  Total = Lang.parseFloat(N)*(r/100+1);
  WMLBrowser.setVar("Sum",Float.int(Total));
  WMLBrowser.refresh();
}
```

Installing a local gateway

This chapter describes how you install and configure a demo version of Ericsson Enterprise WAP Gateway/Proxy 2.0. A local gateway is required for some test configurations as described in *Setting up a WAP application environment* on page 51. It is also required to test push messages.

Note: The demo version of the gateway does not support WTLS. To test WTLS applications you need access to another gateway.

Downloading and installing the gateway

A demo version of Ericsson's Enterprise WAP Gateway/Proxy 2.0 can be downloaded from the Ericsson Developers' Zone (*WAP -> Developers' tools*). Refer to the instructions for installation and configuration. The gateway can be installed on the same machine as WapIDE or on a separate machine.

Configuring the gateway for push

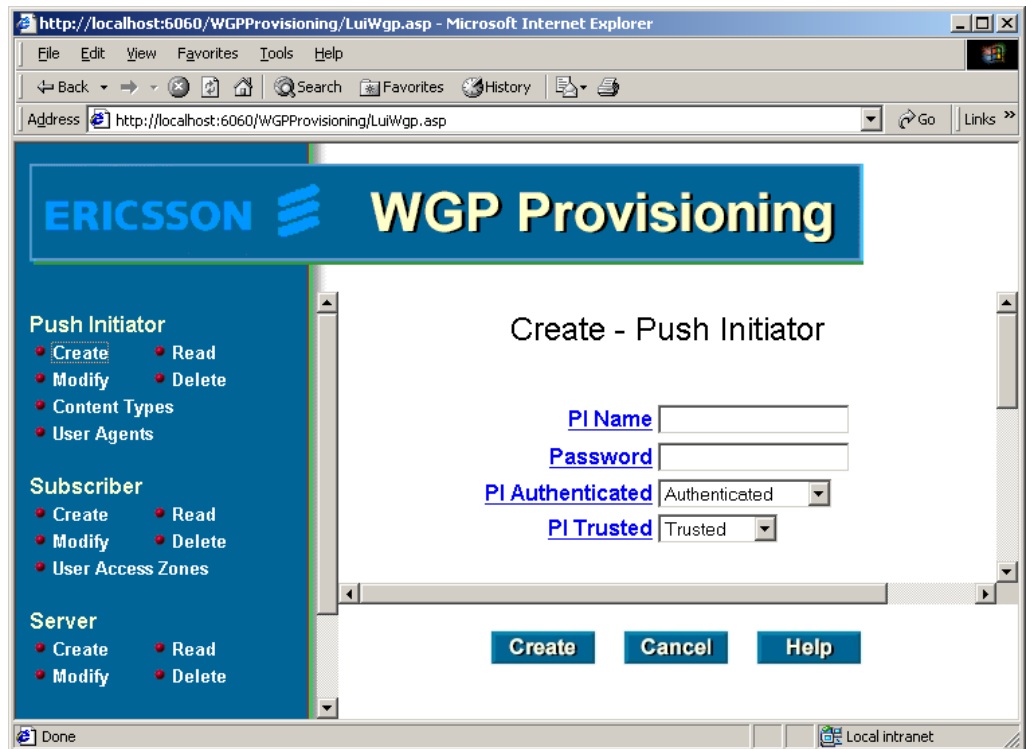
These configuration steps are required to test the WapIDE push functions:

- Define WapIDE as a push initiator.
- Turn off authentication (this is only required if no security is used from the push initiator).
- Define a subscriber to receive the push messages.
- Define receiver IP address (this is necessary to reroute push messages to the WapIDE browser).

Define push initiator

All push initiators must be defined in the gateway.

1. Select **Programs ► Enterprise WGP 2.0 ► EWGP Provisioning** from the Windows Start menu.
2. Press **Push Initiator - Create**.



3. Fill in **PI Name** (e.g. wapide) and **Password** (e.g. Wapide31). The other fields can be left as they are. Press **Create**.

This should create the push initiator in the gateway database and as a Windows user.

If you get the message “WIN2000 account could not be created” the installation is probably not correctly done. Try this procedure:

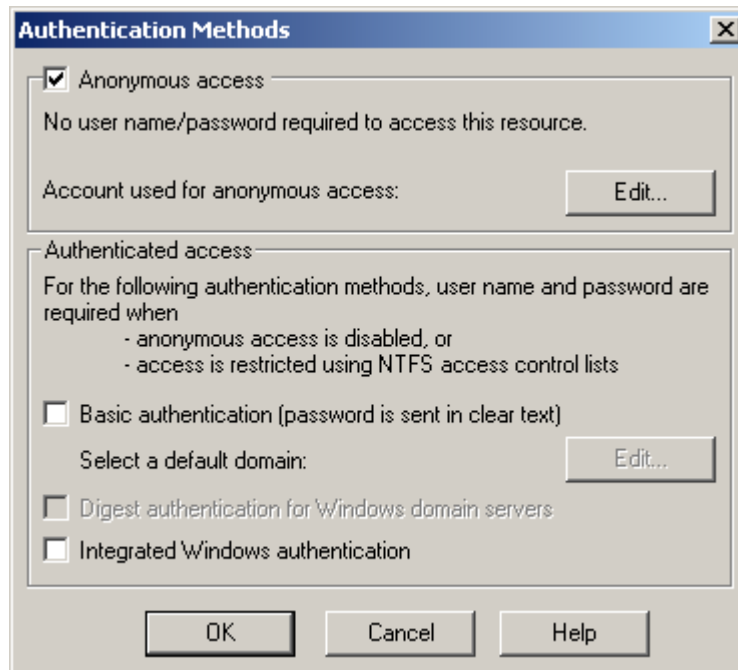
1. Open **Push proxy parameters** as described in *Define receiver IP address* on page 57.
2. In the field **Domain or machine for Push initiators authentication**, insert “\” before the machine name.
3. Restart the push service as described under *Define receiver IP address*.

Then delete and recreate the push initiator (delete may give error messages which you can ignore).

Turn off authentication

Note: This step is only required if you want to use WapIDE (or another push initiator) without HTTP basic authentication security.

1. Select **Programs ► Administrative Tools ► Internet Services Manager** from the Windows Start menu.
2. Select **Default Web Site** and then **ppgctrl** and click on the right mouse button.
3. Select **Properties** from the pop-up menu.
4. Select the **Directory Security** tab.
5. Under **Anonymous access and authentication control** press the **Edit...** button.



6. Activate **Anonymous access**.
7. Press **Edit...** and specify the push initiator created above as anonymous user.
8. Deactivate **Basic authentication**.
9. Press **OK** twice to return to the **Internet Information Services** window.
10. Restart the **Default Web Site**. Select first **Stop** and then **Start** from the pop-up menu.

Define subscriber

One or more subscribers to receive the push messages must be defined in the gateway. When you reroute push messages to the WapIDE browser, it does not matter which subscriber MSISDN number you specify. However, the address (subscriber) you specify when you send push messages must be defined in the gateway.

1. Start **EWGP Provisioning** in the same way as above.
2. Select **Subscriber - Create**.
3. Fill in **MSISDN** (e.g. 11111111), **User ID**, and **Password**. Refer to the online help for further instructions. Press **Create**.

Define receiver IP address

This step is necessary to route push messages to the WapIDE browser. If you are pushing to a real terminal you should not perform this step.

1. Select **Programs ► Enterprise WGP 2.0 ► EWGP Configuration Tool** from the Windows Start menu.
2. Select **Gateway/Proxy** and then **Proxy** and click on the right mouse button.
3. Select **Properties** from the pop-up menu.

4. Select the **Push proxy parameters** tab.

Local Computer

Full proxy parameters | **Push proxy parameters**

Push proxy gateway

Name: myGatewayName

URL: http://www.anUrl.com

Domain or machine for Push initiators authentication (\\ prefix for machine): \\EIP-U030U47AJRH

Bearer gateway to use

Port number: 29002

IP address: 147.214.152.124

Port number for incoming traffic from the bearer gateway: 12948

Max Push messages per second to the bearer gateway: 100

Stack preferences

WDP port number: 2948

Push message delivery preferences

HTTP POST body maximum size (bytes): 100000

Push message database preferences

Push message persistence time after delivery (minutes): 1440

Garbage collection interval (minutes): 60

Number of push messages to deliver before message status update: 1000

Maximum number of push messages to read from database: 1000

Push message queue size: 5000

Retry time after database error occurrence (ms): 60000

Retry time when no push messages in database (ms): 10000

OK Cancel Apply Help

5. Under Bearer gateway to use, enter 29002 for **Port number** and the address of the WapIDE machine for **IP address**. Make a note of the previous values if you want to use the gateway for pushing to real terminals later.

For the changes to take effect, the push service must be restarted:

1. Select **Programs ► Enterprise WGP 2.0 ► EWGP Node supervisor client** from the Windows Start menu.
2. Press **Show Details**.
3. Restart the **Push Delivery Service** by first selecting **Stop Service** and then **Start Service** from the pop-up menu.

Appendix A: Device limitations

The objective of WapIDE is to simulate the real WAP devices as realistically as possible. This chapter describes optional WAP functions not supported by the Ericsson devices (and therefore not by WapIDE), and areas where WapIDE does not work exactly like the real devices.

Devices supported

The following devices are currently supported by WapIDE:

Device	WAP release	Push	UAProf
R320s	1.1		
R380s	1.1		
R520m	1.2.1 ¹	√	√

There are also Chinese versions of these devices (for example, R320sc is the Chinese version of R320s). These versions are partly supported since Chinese characters can be displayed in the device and entered from the computer keyboard. However, Chinese characters can not be entered using the buttons on the device.

Optional WAP features not supported

These optional and new features are not supported (i.e. ignored) by the real devices:

- The **optgroup** (R320s only), **fieldset** (R320s and R520m), and **pre** elements.
- The **align**, **height**, **width**, **hspace**, **vspace**, and **localsrc** attributes on the **image** element (R320s and R520m).
- The **tabindex** attribute on e.g. **select** and **input** elements.
- The **big** and **i** elements on R320s and R520m (displayed as bold and normal font respectively).
- The **i** and **u** elements on R380s (displayed as bold font).

¹ WAP 1.2.1 is also known as the June 2000 conformance release.

Functions not supported by WapIDE

These functions are not supported or work differently in WapIDE.

- **R320s and R520m:**
 - There are no scroll indicators that indicate whether the window can be scrolled up or down or not.
 - In the real devices, when an input window is opened, a text with “abc” or “123” is shown briefly to indicate the format of the input field. WapIDE does not show this text.
 - The maxlength attribute on input fields is not supported.
 - The progress indicator that indicates that a URL is being loaded is not shown in the device display but in the toolbar (rotating globe).
 - Long options are truncated instead of scrolled sideways.
 - The optgroup element is ignored (R520m).
- **R380s:**
 - Long values in input fields are truncated instead of wrapped on multiple lines.
 - The layout of options in a multiple select element is different in some cases. Long options are not wrapped and a right aligned multiple select looks slightly different.
 - Image alignment is different when there are multiple images or buttons with different alignment on the same row.
 - WMLScript confirm dialogs are not scrollable.
 - The hspace and vspace attributes on the image element are not supported.
- **General:**
 - Characters entered in password fields are not shown as asterisks.
 - The soft hyphen character entity (**­** or **­**) is ignored.
 - The maximum size of an image that can be downloaded is different.
 - In Chinese mode, western characters are larger than in the real devices.
 - The WTAI and Crypto WMLScript libraries are not supported.

Glossary of terms

Bookmark	A named reference to a URL. Enables the user to keep track of favorite pages and then return to them, easily, later on.
Card	A single WML unit of navigation and user interface. May contain information to present to the user, instructions for gathering user input, etc.
Character Encoding	Conversion between a sequence of characters and a sequence of bytes and vice versa. Normally, WML document character encoding is captured in transport header attributes such as the Content-Type's "charset" parameter, meta information placed within a document, or the XML declaration.
Client	A device (or application) that initiates a request for connection with a server.
Content	Subject matter (data) stored or generated at an origin server. Content is typically displayed or interpreted by a user agent in response to a user request.
Content Encoding	Conversion of content from one format to another. Typically the resulting format requires less physical space than the original, is easier to process or store, and/or is encrypted. It is can also specify a particular format or encoding standard or process.
Cookie	A cookie is a small piece of information that an HTTP server sends to a browser when the browser connects for the first time. Thereafter, the browser returns a copy of the cookie to the server each time it connects. Typically the server uses the cookie to maintain a session for the user.
Deck	A collection of WML cards. A WML deck is also an XML document.
Device	In this document, used as a synonym for terminal.
DTD	A document type definition (DTD) is a collection of declarations that defines the legal structure, elements, and attributes that are available for use in an XML document that complies with the DTD.
HTTP	The Hypertext Transfer Protocol is the client/server protocol that defines how messages are formatted and transmitted on the World Wide Web.
JavaScript	A <i>de facto</i> standard language that can be used to add dynamic behaviour to HTML documents. JavaScript is one of the

originating technologies of ECMAScript.

Origin Server	The server on which a given resource resides or is to be created. Often referred to as a web server or an HTTP server.
Proxy Server	An HTTP server, typically running on a firewall machine, that provides access to the outside world for clients on a local network.
Resource	A network data object or service that can be identified by a URL.
Terminal	A device providing the user with user agent capabilities, including the ability to request and receive information. Also called a mobile terminal or mobile station.
UAProf	User agent profiles specify device characteristics that are used by the origin server for content formatting purposes.
Unicode	A universal character encoding scheme for written characters and text. It defines a consistent way of encoding multilingual text that enables the exchange of text data internationally. Unicode provides for two encoding forms: a default 16-bit form and a byte-oriented form called UTF-8.
URL	A Uniform Resource Locator is an address identifying the location of a file on the Internet, consisting of the protocol, the computer on which the file is located, and the file's location on that computer.
User Agent	A user agent is any software or device that interprets WML, WMLScript, WTAI or other resources. This may include textual browsers, voice browsers, search engines, etc.
WAP Gateway	A WAP gateway converts between the WAP protocol stack (WSP, WTP, and WDP) and the WWW protocol stack (HTTP and TCP/IP), and performs encoding of WML and WMLScript.
WapIDE	This is an acronym for Ericsson's Wireless Application Protocol Integrated Development Environment.
WML	The Wireless Markup Language is a hypertext markup language used to represent information for delivery to a narrowband device, e.g., a phone.
WMLScript	A scripting language used to program the mobile device. WMLScript is an extended subset of the JavaScript scripting language.
WTA	Wireless Telephony Application. A framework for accessing the telephony related functions in a mobile terminal.
WTLS	Wireless Transport Layer Security. A security protocol based upon the industry-standard Transport Layer Security (TLS) protocol, formerly known as Secure Sockets Layer (SSL).
XML	The Extensible Markup Language is a World Wide Web Consortium (W3C) standard for Internet markup languages, of which WML is one such language.