Developers guidelines

DEVELOPER WORLD THE FAST TRACK FROM MIND TO MARKET

March 2009

NetFront™ v3.3–3.4 Web browser

in Sony Ericsson phones

Sony Ericsson

Preface

Purpose of this document

This document describes how to develop Web pages for the browsers in the phones listed under “Products” on page 3.

The document is intended for content developers for Sony Ericsson phones. The main purpose is not to instruct the reader how to write HTML and XHTML content, but to describe how the phone browser interprets tags and attributes.

People who may benefit from this document are:

- Content developers
- Operators and service providers
- Content providers.

It is assumed that the reader is familiar with the markup languages HTML and XHTML, and the style sheet language CSS.

These Developers guidelines are published by:

Sony Ericsson Mobile Communications AB,
SE-221 88 Lund, Sweden

www.sonyericsson.com/

25th edition (March 2009)
Publication number: EN/LZT 108 8099 R25A

Sony Ericsson Developer World

At www.sonyericsson.com/developer, developers find the latest technical documentation and development tools such as phone White papers, Developers guidelines for different technologies, Getting started tutorials, SDKs (Software Development Kits) and tool plugins. The Web site also features news articles, go-to-market advice, moderated discussion forums offering free technical support and a Wiki community sharing expertise and code examples.

For more information about these professional services, go to the Sony Ericsson Developer World Web site.

Document conventions

Products

Sony Ericsson mobile phones are referred to in this document using generic names. The table also lists the NetFront browser version, which phones support Web feeds (RSS), and which phones emulate mouse cursors in the browser.

Generic names Series	**Sony Ericsson mobile phones**	**Browser version**	**Web feed support**	**Mouse cursor UI**
C510	C510, C510c, C510a	3.4	Yes	Yes
C702	C702, C702c, C702a	3.4	Yes	Yes
C901	C901, C901a	3.4	Yes	Yes
C902	C902, C902c	3.4	Yes	Yes
C903	C903, C903a	3.4	Yes	Yes
C905	C905, C905c, C905a	3.4	Yes	Yes
G502	G502, G502c	3.4	Yes	Yes
G705	G705, G705u	3.4	Yes	Yes
K310	K310i, K310c, K310a	3.3		
K320	K320i, K320c	3.3		
K510	K510i, K510c	3.3		
K530	K530i	3.3	Yes	
K550	K550i, K550c	3.3	Yes	
K630	K630i	3.4	Yes	

K660	K660i	3.4	Yes	Yes
K770	K770i	3.3	Yes	
K790	K790i, K790c, K790a	3.3	Yes	
K800	K800i, K800c	3.3	Yes	
K810	K810i, K818c	3.3	Yes	
K850	K850i, K858c	3.4	Yes	
S500	S500i, S500c	3.3	Yes	
T650	T650i, T658c	3.3	Yes	
T700	T700	3.4	Yes	Yes
T707	T707, T707a	3.4	Yes	Yes
V640	V640i	3.4	Yes	
W200	W200i, W200c	3.3		
W300	W300i, W300c	3.3		
W350	W350i, W350c	3.3	Yes	
W380	W380i, W380c	3.3	Yes	
W508	W508, W508c, W508a, W518a	3.4	Yes	Yes
W550	W550i, W550c	3.3		
W580	W580i, W580c	3.3	Yes	
W595	W595, W595s	3.4	Yes	Yes
W600	W600i	3.3		
W610	W610i, W610c	3.3	Yes	
W660	W660i	3.3	Yes	
W705	W705, W705u	3.4	Yes	Yes
W710	W710i, W710c	3.3	Yes	
W715	W715	3.4	Yes	Yes
W760	W760i, W760c	3.4	Yes	Yes
W810	W810i, W810c, W810a	3.3		
W830	W830i, W830c	3.3	Yes	
W850	W850i, W850c	3.3	Yes	
W880	W880i, W888c	3.3	Yes	
W890	W890i	3.4	Yes	Yes
W900	W900i	3.3		
W902	W902	3.4	Yes	Yes

W910	W910i, W908c	3.4	Yes	
W980	W980i	3.4	Yes	Yes
W995	W995, W995a	3.4	Yes	Yes
Z310	Z310i, Z310a	3.3	Yes	
Z530	Z530i, Z530c	3.3		
Z550	Z550i, Z550c, Z550a	3.3		
Z555	Z555i, Z555a	3.3	Yes	
Z558	Z558i, Z558c	3.3		
Z610	Z610i	3.3	Yes	
Z710	Z710i, Z710c	3.3	Yes	
Z750	Z750i	3.4	Yes	
Z770	Z770i	3.4	Yes	Yes
Z780	Z780i, Z780a	3.4	Yes	Yes

Terminology

AJAX Asynchronous JavaScript and XML

A web development technique for creating interactive web applications. By exchanging small amounts of data with the server “behind the scene”, the entire web page does not have to be reloaded each time the user requests a change.

CSS Cascading Style Sheets

De facto standard style sheet language, defined by the W3C and supported by IE, Netscape and Opera

HTML Hypertext Markup Language

The original Web markup language, supported by IE, Netscape and Opera. Most Web pages are HTML documents

I-mode HTML An extended subset of HTML, defined by NTT DoCoMo

OMA Open Mobile Alliance

The primary standardisation organisation for mobile applications.
See http://www.openmobilealliance.org/

RSS Really Simple Syndication

A format for Web feeds

W3C The World Wide Web Consortium

	The primary standardisation organisation on the Web. See http://www.w3.org/
WCSS	Wireless CSS An extended subset of CSS defined by the OMA
XHTML	Extensible Hypertext Markup Language The latest version of HTML is called XHTML, since it is based on XML. On a functional level, HTML and XHTML are essentially the same. On a syntactical level, XHTML requires the author to be more careful with syntax. The stricter syntax which XHTML inherits from XML, makes the code easier to process
XHTML Basic	A subset of XHTML defined by the W3C
XHTML Mobile profile	A subset of XHTML, a superset of XHTML Basic, defined by OMA
XML	A set of syntactical rules for markup languages. All new markup languages, such as Scalable Vector Graphics (SVG) and Synchronized Multimedia Integration Language (SMIL) are based on XML

Typographical conventions

The following typographical conventions are used in this document.

HTML element names are written inside “<“ and “>”:
<html>

Code is written in Courier font:
`<html>..</html>`

Trademarks and acknowledgements

Document history

Change history		
2005-10-17	Version R1A	First edition. Information about the W600, W550 and W900 series
2005-10-25	Version R1B	Minor editorial changes
2006-01-04	Version R2A	Second edition. Information added about the W810 series
2006-02-13	Version R3A	Third edition. Information added about the K610 series
2006-02-28	Version R4A	Fourth edition. Information added about the K800, K790, Z530, W300, K510 and K310 series
2006-05-19	Version R5A	Fifth edition. Information added about the Z550, W850, Z710 and W710 series
2006-06-12	Version R5B	Fifth revised edition
2006-08-22	Version R6A	Sixth edition. Information added about the K618i and the Z610 series
2006-09-25	Version R7A	Seventh edition. Information added about W830, K320 and Z558 series and the Z550a
2006-12-04	Version R8A	Eighth edition. Information added about Z310 series
2007-01-08	Version R9A	Ninth edition. Information added about W200 series
2007-02-06	Version R10A	10th edition. Information added about W880, K550, W610 and K810 series
2007-03-13	Version R11A	11th edition. Information added about W660 series
2007-05-08	Version R12A	12th edition. Information added about S500, T650, W580 and Z750 series. NetFront version 3.4 and RSS information added
2007-06-14	Version R13A	13th edition. Information added about K530, K850 and W910 series
2007-08-21	Version R14A	14th edition. Information added about K770 series
2007-09-21	Version R15A	15th edition. Information added about V640 series
2007-11-06	Version R16A	16th edition. Information added about K630, K660, W380 and W890 series

Change history

2008-02-20	Version R17A	17th edition. Information added about C702, C902, W350, W760, W980, Z555 and Z770 series
2008-05-20	Version R18A	18th edition. Information added about G502 and Z780 series
2008-06-17	Version R19A	19th edition. Information added about C905 series
2008-07-22	Version R20A	20th edition. Information added about T700, W595 and W902 series
2008-09-09	Version R21A	21st edition. Information added about G705 series
2008-10-17	Version R21B	21st revised edition. New document layout
2008-11-11	Version R22A	22nd edition. Information added about W705 series
2009-01-09	Version R23A	23rd edition. Information added about C510 and W508 series
2009-02-15	Version R24A	24th edition. Information added about C901, C903, W715 and W995 series
2009-03-26	Version R25A	25th edition. Information added about T707 series

Contents

Overview

The Web browser is a development platform that lets content providers create content with standard Web technologies such as HTML (Hypertext Markup Language) and the style sheet language CSS (Cascading Style Sheets).

Technology and architecture

Multimedia download services (for example download of ringtones and video clips) have become an increasingly important category of Web-based services. A separate section of this document is devoted to download services and the use of the OMA Download standard.

NetFront™ v3.3 and v3.4 browser (NF3)

NetFront™ v3.3 and v3.4 browser engines, utilised in the Sony Ericsson phone models within the scope of this document, includes support for WAP 2.0, WML, HTML 4.01, XHTML and CSS2. The browser uses Rapid-Render™, which is an incremental rendering technology that significantly accelerates the presentation of Web pages, and Smart-Fit Rendering™, a rendering technology that intelligently renders Web pages to fit the screen width of mobile devices, eliminating the need for horisontal scrolling.

Column Rendering is a rendering mode implemented in phones with mouse cursor interface emulation. When Smart-Fit is deactivated in these phones, pages are rendered using this method. Without Column Rendering, text exceeding the screen width requires repeated use of the scroll bar. With Column Rendering this issue has been resolved:

- Text content is rendered to fit the width of the screen using a line break function.
- Column Rendering renders almost exactly what you would see on a PC, also supporting vertical scrolling.

Browser compliancy

The Web browser is compliant with the following standards:

- HTML ver. 4.01, www.w3.org/TR/html401
- XHTML Basic 1.0, http://www.w3.org/TR/xhtml-basic
- XHTML 1.1, Module-based XHTML, http://www.w3.org/TR/xhtml11
- XHTML Mobile profile, http://www.openmobilealliance.org/

- Compact HTML for Small Information Appliances, http://www.w3.org/TR/1998/NOTE-compactHTML-19980209
- WML 1.3, http://www.openmobilealliance.org/
- Cascading Style Sheets, level 2 revision 1, CSS 2.1 Specification, http://www.w3.org/TR/CSS21
- ECMA-script according to specifications found at http://www.ecma-international.org/publications/standards/Stnindex.htm
- ECMAScript Mobile Profile, <http://www.openmobilealliance.org/>
- Document Object Model Level 2 (DOM 2), http://www.w3.org/DOM/DOMTR#dom2
- Scalable Vector Graphics, Mobile SVG Profile: SVG Tiny, version 1.1+ http://www.w3.org/TR/SVGMobile/
- Web applications developed according to the AJAX development principles are supported. The XMLHTTPRequest standard which is used for AJAX can be found at http://www.w3.org/TR/XMLHttpRequest/

Limitations

Font limitations

The phones contain one proportional sans-serif font. In most phones the sans-serif font is scalable to any font size. Bold, Italic, Underlined and Strikeout font styles are supported.

Note: In the W350, W380, Z310 and Z555 series, font sizes are limited. For Western languages three font sizes are available, “small”, “medium” and “large”. For Chinese and Japanese there are two sizes; “medium” and “large” give the same font size in the browser.

Frames

Frames are only supported in the 3.4 version of the browser. In the 3.3 version of the browser, HTML inside the <noframes> element is displayed.

HTML support

This section lists all HTML elements and related attributes supported by the Sony Ericsson phones that use the NetFront browser, versions 3.3 and 3.4. Some HTML elements that are not supported are also included in the list. The list only contains supported attributes.

For detailed information about the HTML syntax, permitted attribute values, and so on, see information found on www.w3.org.

Content vs design

Separation between the content (text, images, table, forms, and so on) and presentation (colour, fonts, borders, layout, and so on) is a fundamental design rule on the Web. On a Web page, content is described by the markup language HTML while presentation is described by the style sheet language CSS.

The same HTML document can be presented in different ways by adding different style sheets, and the same style sheet can be used for many documents. Both HTML and CSS are de facto Web standards, supported by all major Web browsers on the market, and are familiar to Web developers.

Common attributes

Some attributes are defined for a large number of elements and will be listed as “Common attributes” below. Although the common attributes are defined for many elements, they may not always be useful to all of them. For example, applying a style rule to the <head> element is not useful, because the head will not be displayed.

The common attributes are:

- `class`
- `dir`
- `id`
- `title`
- `lang`
- `style`
- `onkeypress`
- `onkeydown`
- `onkeyup`

Note: `onkeypress`, `onkeydown` and `onkeyup` are supported for all elements. However, they are normally only used on selectable objects like the <a> link, form <input> elements, and the <body> element.

Note: The browser supports the text direction attribute with values dir="ltr" and dir="rtl". In order to display different languages used on the same page (nested) the browser uses a special algorithm called "The Bidirectional Algorithm" provided by Unicode (http://unicode.org/reports/tr9/).

Mouse events

The following mouse events are supported in phones implementing the mouse cursor interface in the browser. In all other phones these events are not supported. In the tables below, “Mouse events” in the attribute column indicates that the following events are supported:

- `ondblclick`
- `onmousedown`
- `onmouseout`
- `onmouseup`
- `onmouseover`
- `onmousemove`

The following mouse event is **not** supported in any Sony Ericsson phone.

- `onmousewheel`

Elements and attributes

Element

	attribute	Remarks
<a>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`accesskey`	After the execution, an event handler of control is initialised
	`coords`	
	`href`	NF3 supports jumping via fragment identifier only for <a> and <img >elements
	`name`	
	`onblur`	
	`onfocus`	
	`shape`	Only rectangle focus frames are supported
	`tabindex`	
	`target`	

Element / attribute	Remarks
<abbr>	
Common attributes	
Mouse events	**Only** in phones with mouse cursor support
<acronym>	
Common attributes	
Mouse events	**Only** in phones with mouse cursor support
<address>	
Common attributes	
Mouse events	**Only** in phones with mouse cursor support
<applet>	**Not supported**
<area>	
Common attributes	
Mouse events	**Only** in phones with mouse cursor support
`accesskey`	After the execution, an event handler of control is initialised
`coords`	
`href`	
`nohref`	
`onblur`	
`onclick`	Triggered after text has been entered in the control
`onfocus`	
`shape`	Only rectangle focus frames are supported
`tabindex`	
`target`	
<b>	
Common attributes	
Mouse events	**Only** in phones with mouse cursor support
<base>	
`href`	
`target`	

Element attribute	Remarks
<basefont>	The <basefont> element does not change the font size. A value for "color" attribute takes effect only when it is defined in the <head> element
Common attributes	
color	
size	
<bgsound>	**Note:** When the autostart attribute is set to “false”, the sound is not played. When set to “true”, the sound starts to play as soon as the page has been rendered completely.
Common attributes	
balance	
loop	
src	
volume	
<big>	
Common attributes	
Mouse events	**Only** in phones with mouse cursor support
<blink>	
<blockquote>	
Common attributes	
Mouse events	**Only** in phones with mouse cursor support
cite	
<body>	
Common attributes	
Mouse events	**Only** in phones with mouse cursor support
alink	
background	
bgcolor	
bgproperties	
bottommargin	
leftmargin	
link	

Element	attribute	Remarks
	marginheight	
	marginwidth	
	onload	
	onunload	
	rightmargin	
	scroll	
	text	
	topmargin	
	vlink	
	Common attributes	
	clear	
<button>		Label string specified in the <button> element is displayed top-aligned
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	accesskey	After the execution, an event handler of control is initialised
	disabled	Depending on graphic system
	name	
	onblur	
	onclick	Mapped to the Selection key
	onfocus	
	tabindex	
	type	
	value	
<caption>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support

Element	attribute	Remarks
	`align`	If left is specified, the vertical coordinate is the same as top and the horisontal coordinate is left-aligned in the table width. If right is specified, the vertical coordinate is the same as top and the horisontal coordinate is right-aligned in the table width
<cite>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
<code>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
<col>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`bgcolor`	
	`span`	
	`valign`	
	`width`	
<colgroup>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`bgcolor`	
	`span`	
	`valign`	
	`width`	
<comment>	Not supported	
<dd>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
<del>		

Element	attribute	Remarks
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`cite`	There is no visual effect even if the attribute value is specified
<dfn>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
<dir>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`align`	"justify" is not supported for this attribute
<dl>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
<dt>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
<embed>		
	`align`	**Ver 3.4 only**
	`alt`	**Ver 3.4 only**. Whether it affects behaviour of the attribute value depends on the plugin implementation
	`border`	**Ver 3.4 only**
	`code`	**Ver 3.4 only**. Whether it affects behaviour of the attribute value depends on the plugin implementation
	`codebase`	**Ver 3.4 only**. Whether it affects behaviour of the attribute value depends on the plugin implementation

Element

attribute	Remarks
frameborder	**Ver 3.4 only**
height	
hidden	
hspace	
name	
palette	**Ver 3.4 only**. Whether it affects behaviour of the attribute value depends on the plugin implementation
pluginspage	**Ver 3.4 only**. Whether it affects behaviour of the attribute value depends on the plugin implementation
pluginurl	**Ver 3.4 only**. Whether it affects behaviour of the attribute value depends on the plugin implementation
src	
type	
units	**Ver 3.4 only**. Whether it affects behaviour of the attribute value depends on the plugin implementation
vspace	
width	
<fieldset>	
Common attributes	
Mouse events	**Only** in phones with mouse cursor support
<font>	
Common attributes	
color	
size	Only “small”, “medium” and “large” font sizes are supported in **W350, W380, Z310** and **Z555** series
<form>	
Common attributes	
Mouse events	**Only** in phones with mouse cursor support
accept	
accept-charset	
action	NF3 submits only when "action" is specified
enctype	

Element	attribute	Remarks
	method	
	name	
	onclick	Triggered after text has been entered in the control
	onreset	
	onsubmit	
	target	
<frame>		Ver 3.4 only
	Common attributes	Ver 3.4 only
	frameborder	Ver 3.4 only
	marginheight	Ver 3.4 only
	marginwidth	Ver 3.4 only
	name	Ver 3.4 only
	noresize	Ver 3.4 only
	onload	Ver 3.4 only
	scrolling	Ver 3.4 only
	src	Ver 3.4 only
<frameset>		Ver 3.4 only
	border	Ver 3.4 only
	cols	Ver 3.4 only
	frameborder	Ver 3.4 only
	onload	Ver 3.4 only
	onunload	Ver 3.4 only
	rows	Ver 3.4 only
<h1>, <h2>, ..., <h6>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	align	"justify" is not supported for this attribute
<head>		

Element	attribute	Remarks
	Common attributes	
	`lang`	The browser internally administers this attribute value. There is no visual effect even if the attribute value is specified
<hr>		NF3 does not apply the text-align property setting to <hr> elements
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`align`	
	`color`	
	`noshade`	
	`size`	
	`width`	
<html>		
	`fragments`	
	`lang`	The browser internally administers this attribute value. There is no visual effect even if the attribute value is specified
	`version`	
<i>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
<iframe>		Supported in C510, C702, C901, C902, C903, C905, G502, G705, K530, K550, K630, K660, K770, K810, K850, S500, T650, T700, T707, V640, W508, W580, W595, W610, W660, W705, W710, W715, W760, W830, W850, W880, W890, W902, W910, W980, W995, Z610, Z710, Z750, Z770 and Z780 series and in later production series of the K610, K790 and K800. Early K610, K790 and K800 phones do not support <iframe>, but may do so after updating the phone software to the latest available version
	Common attributes	
	`align`	
	`frameborder`	
	`height`	
	`marginheight`	
	`marginwidth`	

Element	attribute	Remarks
	`name`	
	`scrolling`	
	`src`	
	`width`	
<ilayer>		Not supported.
<image>		NF3 processes <image> elements in the same way as it does <img> elements
<img>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`align`	
	`alt`	
	`border`	
	`height`	
	`hspace`	
	`ismap`	
	`name`	
	`onload`	
	`src`	
	`usemap`	
	`width`	
	`vspace`	
<input>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`accesskey`	After the execution, an event handler of control is initialised
	`disabled`	
	`onblur`	
	`onchange`	
	`onclick`	Mapped to Selection key
	`onfocus`	

Element	attribute	Remarks
	tabindex	
<input type = “button”>		
	name	
	value	
<input type = “checkbox”>		
	checked	
	name	
	value	
<input type = “file”>		
	accept	
	name	
	size	
	value	
<input type = “hidden”>		
	name	
	value	
<input type = “image”>		
	align	
	alt	
	border	
	height	
	ismap	
	name	
	src	
	usemap	
	value	
	width	
<input type = “password”>		
	maxlength	
	name	

Element	attribute	Remarks
	size	
	value	
	readonly	
<input type = “radio”>		
	checked	
	name	
	value	
<input type = “reset”>		
	name	
	value	
<input type = “submit”>		
	name	
	value	
<input type = “text”>		
	maxlength	
	name	
	size	
	value	
	readonly	
<ins>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	cite	There is no visual effect even if the attribute value is specified
<isindex>		Not supported
<kbd>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
<keygen>		Not supported
<label>		

Element	attribute	Remarks
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`accesskey`	After the execution, an event handler of control is initialised
	`for`	
<layer>		Not supported
<legend>		Not supported
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`onclick`	Triggered after text has been entered in the control
	`type`	
	`value`	
<link>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`href`	
	`media`	Partial support for style sheet
	`rel`	Only `rel = ”stylesheet"` is supported
	`type`	`type = "text/css"` is supported. MIME types for Web feeds are also supported with this attribute
<listing>		Not supported
<map>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`name`	
<marquee>		
	`behavior`	
	`bgcolor`	
	`direction`	
	`height`	

Element	attribute	Remarks
	hspace	
	loop	When this attribute is omitted and the behavior attribute is set to “slide” or “display”: • If -wap-marquee is specified for the style attribute of the marquee element, scrolling occurs once. • In other cases, scrolling occurs infinitely
	scrollamount	
	scrolldelay	
	width	
	vspace	
<menu>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
<meta>		
	content	Partial support. Content in the <meta> element is supported only if refresh, expires, pragma, cache-control, set-cookie are set for http-equiv
	http-equiv	
	name	NF3 can parse and retrieve the value of the name attribute in the content
	type	
<nextid>		Not supported
<nobr>		Not supported
<noembed>		
<noframes>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
<nolayer>		Not supported
<noscript>		
	Common attributes	

Element

attribute	Remarks
<object>	
Common attributes	
Mouse events	**Only** in phones with mouse cursor support
`accesskey`	**Ver 3.4 only**. Whether it affects behaviour of the attribute value depends on the plugin implementation
`align`	
`archive`	**Ver 3.4 only**. Whether it affects behaviour of the attribute value depends on the plugin implementation
`border`	
`classid`	
`code`	**Ver 3.4 only**. Whether it affects behaviour of the attribute value depends on the plugin implementation
`codebase`	**Ver 3.4 only**. Whether it affects behaviour of the attribute value depends on the plugin implementation
`codetype`	**Ver 3.4 only**. Whether it affects behaviour of the attribute value depends on the plugin implementation
`data`	
`declare`	**Ver 3.4 only**. Whether it affects behaviour of the attribute value depends on the plugin implementation
`height`	If a fractional value without % is specified, the browser uses the integer part of the value. The browser supports fractional values with %, for example, `height="1.2%"`, and also values starting with a decimal point, for example, `height=".2%"`.
`hspace`	
`name`	
`standby`	**Ver 3.4 only**. Whether it affects behaviour of the attribute value depends on the plugin implementation
`type`	
`usemap`	
`width`	If a fractional value without % is specified, the browser uses the integer part of the value. The browser supports fractional values with %, for example, `width="1.2%"`, and also values starting with a decimal point, for example, `width=".2%"`.
`vspace`	

Element	attribute	Remarks
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`start`	
	`type`	
<optgroup>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`disabled`	
	`label`	
	`onclick`	Mapped to the Selection key
<option>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`disabled`	
	`label`	
	`onclick`	Mapped to the Selection key
	`selected`	
	`style`	Available CSS properties: `color`, `background-color`
	`value`	
<p>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`align`	"justify" is not supported for this attribute
<param>		
	`name`	NF3 supports only <param> elements specified between the `start` and `end` tags of an <object> element
	`value`	NF3 supports only <param> elements specified between the `start` and `end` tags of an <object> element
<plaintext>		

Element	attribute	Remarks
<pre>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
<q>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`cite`	
<rb>		Not supported
<rbc>		Not supported
<rp>		Not supported
<rt>		Not supported
<rtc>		Not supported
<ruby>		Not supported
<s>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
<samp>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
<script>		
	`charset`	
	`defer`	
	`src`	
	`type`	
<select>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`accesskey`	NF3 specific. After the execution, an event handler of control is initialised
	`disabled`	

Element	attribute	Remarks
	`multiple`	
	`name`	
	`onblur`	"label" is not supported
	`onchange`	
	`onclick`	Mapped to the Selection key
	`onfocus`	"label" is not supported
	`tabindex`	
<small>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
<strike>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
<style>		
	Common attributes	
	`media`	
	`type`	
<sub>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
<sup>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support

Element	attribute	Remarks
<table>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`align`	
	`background`	
	`bgcolor`	
	`border`	
	`bordercolor`	
	`cellpadding`	
	`cellspacing`	
	`frame`	
	`height`	
	`rules`	
	`width`	
<tbody>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`align`	Supported values: “left”, “center”, “right”. Not supported values: "justify", "char"
	`bgcolor`	
	`valign`	Supported values: “top”, “middle”, “bottom”, “baseline”
<td>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`align`	Supported values: “left”, “center”, “right”. Not supported values: "justify", "char"
	`background`	
	`bgcolor`	
	`bordercolor`	
	`colspan`	
	`height`	

Element	attribute	Remarks
	nowrap	
	rowspan	
	valign	Supported values: “top”, “middle”, “bottom”, “baseline”
	width	
<textarea>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	accesskey	After the execution, an event handler of control is initialised
	cols	
	disabled	
	name	
	onblur	
	onchange	
	onclick	Triggered after text has been entered in the control
	onfocus	
	readonly	
	rows	
	tabindex	
<tfoot>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	align	Supported values: “left”, “center”, “right”. Not supported values: "justify", "char"
	bgcolor	
	valign	Supported values: “top”, “middle”, “bottom”, “baseline”
<th>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	align	Supported values: “left”, “center”, “right”. Not supported values: "justify", "char"
	background	

Element	attribute	Remarks
	bgcolor	
	bordercolor	
	colspan	
	height	
	nowrap	
	rowspan	
	valign	Supported values: “top”, “middle”, “bottom”, “baseline”
	width	
<thead>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	align	Supported values: “left”, “center”, “right”. Not supported values: "justify", "char"
	bgcolor	
	valign	Supported values: “top”, “middle”, “bottom”, “baseline”
<title>		Ignored element.
<tr>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	align	Supported values: “left”, “center”, “right”. Not supported values: "justify", "char"
	background	
	bgcolor	
	bordercolor	
	height	
	valign	Supported values: “top”, “middle”, “bottom”, “baseline”
<tt>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
<u>		

Element	attribute	Remarks
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
	`type`	
<var>		
	Common attributes	
	Mouse events	**Only** in phones with mouse cursor support
<wbr>	Not supported	
<xmp>		

Cascading Style Sheets

Before style sheets were introduced on the Web, developers had little control over the presentation of their Web pages. An XHTML document specifies the structure of the content; which part is a paragraph, which part is a heading, and so on. It does not specify how content should be presented. Browsers use a default presentation for documents without style sheets. By adding a style sheet to the document, the developer can control the presentation of the document: colours, fonts and layout.

CSS

On the Web, the de facto standard style sheet language is Cascading Style Sheets (CSS), specified by the W3C and implemented in all major Web browsers on the market. For mobile phones, the OMA has identified a subset of CSS and added OMA-specific style rules to it. The CSS subset and the OMA extensions are called Wireless CSS (WCSS).

The NetFront v3.3 – v3.4 (NF3) browsers conform to the following CSS standards:

- Cascading Style Sheets, level 1 (CSS 1 Specification) available at http://www.w3.org/TR/REC-CSS1
- Cascading Style Sheets, level 2 (CSS 2 Specification) available at http://www.w3.org/TR/REC-CSS2
- CSS Mobile Profile 1.0, available at http://www.w3.org/TR/CSS-mobile
- Specification Information Notes, WAP-239-101-WCSS-20020430-a.pdf and WAP-239-WCSS-20011026-a.pdf, available at http://www.openmobilealliance.org/release_program/browsing_archive.html.

A style sheet is a collection of style rules. A style rule is a statement about some presentational characteristic of an element. A selector binds the style rule to an element.

The CSS MIME type

When a WCSS style sheet is delivered to the browser the MIME type is “text/css”.

Selectors

The selector acts as some kind of search criteria. All elements in the document that match the criteria are selected. A selector may select all elements with a certain name, all elements which are descendants of a specific element, or an element with a specific identifier.

Supported selector patterns

Pattern	Remarks
*	
E	
E F	
E > F	
E:first-child	
E:link	Available HTML element: <a>
E:visited	Available HTML element: <a>
E:active	An event occurs after mouseup. Available HTML elements: <a>, <img> (when the `ismap` attribute is specified)
E:hover	Available HTML element: <a>
E:focus	Available HTML elements: <a>, <input>, <textarea>, <select>, <button>, <embed>, <iframe>
E:lang(c)	
E + F	
E[foo]	
E[foo="warning"]	
E[foo~="warning"]	
E[lang|="en"]	
DIV.warning	
E#myid	

Pattern	Remarks
E:before	HTML elements to which property value is **not** applied: <html>, <head>, <meta>, <link>, <script>, <style>, <base>, <frameset>, <frame>, <iframe>, <img>, <embed>, <object>, <param>, <input>, <textarea>, <button>, <select>, <option>, <opt-group>
E:after	HTML elements to which property value is **not** applied: <html>, <head>, <meta>, <link>, <script>, <style>, <base>, <frameset>, <frame>, <iframe>, <img>, <embed>, <object>, <param>, <input>, <textarea>, <button>, <select>, <option>, <opt-group>
E, F	

CSS properties

The following tables list supported CSS properties, grouped by types of elements to which they apply.

Boxes

Property	Value	Media group	Remarks
margin-top		visual	Not for <input> elements
	<length>		
	<percentage>		
	auto		
	inherit		
margin-right		visual	Not for <input> elements
	<length>		
	<percentage>		
	auto		
	inherit		
margin-bottom		visual	Not for <input> elements

Property	Value	Media group	Remarks
	<length>		
	<percentage>		
	auto		
	inherit		
margin-left		visual	Not for <input> elements
	<length>		
	<percentage>		
	auto		
	inherit		
margin		visual	Not for <input> elements. If the `margin` property is specified in the element stated between the `start` and `end` tags of the <td> element, the area specified by the element is displayed and no margin is left at the top and bottom of the area
	<margin-width>{1,4}		
	inherit		
padding-top		visual	
	<length>		
	<percentage>		
	inherit		
padding-right		visual	
	<length>		
	<percentage>		
	inherit		
padding-bottom		visual	
	<length>		
	<percentage>		
	inherit		
padding-left		visual	
	<length>		

Property	Value	Media group	Remarks
	<percentage>		
	inherit		
padding		visual	If the `padding` property is specified for the control defined by the <input> (except type="hidden"), <select>, <textarea> and <button> elements, padding is set outside the control
	<padding-width>{1,4}		
	inherit		
border-top-width		visual	
	thin		
	medium		
	thick		
	<length>		
	inherit		
border-right-width		visual	
	thin		
	medium		
	thick		
	<length>		
	inherit		
border-bottom-width		visual	
	thin		
	medium		
	thick		
	<length>		
	inherit		
border-left-width		visual	
	thin		
	medium		
	thick		

Property	Value	Media group	Remarks
	<length>		
	inherit		
border-width		visual	
	<border-width>{1,4}		
	inherit		
border-top-color		visual	
	<color>		
	transparent		NF3-specific
	inherit		
border-right-color		visual	
	<color>		
	transparent		NF3-specific
	inherit		
border-bottom-color		visual	
	<color>		
	transparent		NF3-specific
	inherit		
border-left-color		visual	
	<color>		
	transparent		NF3-specific
	inherit		
border-color		visual	
	<color>{1,4}		
	transparent		NF3-specific
	inherit		
border-top-style		visual	
	none		
	hidden		
	dotted		
	dashed		

Property	Value	Media group	Remarks
	solid		
	double		
	groove		
	ridge		
	inset		
	outset		
	inherit		
border-right-style		visual	
	none		
	hidden		
	dotted		
	dashed		
	solid		
	double		
	groove		
	ridge		
	inset		
	outset		
	inherit		
border-bottom-style		visual	
	none		
	hidden		
	dotted		
	dashed		
	solid		
	double		
	groove		
	ridge		
	inset		
	outset		
	inherit		

Property	Value	Media group	Remarks
border-left-style		visual	
	none		
	hidden		
	dotted		
	dashed		
	solid		
	double		
	groove		
	ridge		
	inset		
	outset		
	inherit		
border-style		visual	The `border-style` property setting is not applied to the parts of a form that are defined by <input>, <select>, <option> and <textarea> elements
	<border-style>{1,4}		
	inherit		
border-top		visual	The `border-top` property setting is not applied to the parts of a form that are defined by <input>, <select>, <option> and <textarea> elements
	'border-top-width' || 'border-top-style' || 'border-top-color		
	inherit		
border-right		visual	The `border-right` property setting is not applied to the parts of a form that are defined by <input>, <select>, <option> and <textarea> elements
	'border-right-width' || 'border-right-style' || 'border-right-color		
	inherit		

Property	Value	Media group	Remarks
border-bottom		visual	The `border-bottom` property setting is not applied to the parts of a form that are defined by <input>, <select>, <option> and <textarea> elements
	'border-bottom-width' || 'border-bottom-style' || 'border-bottom-color		
	inherit		
border-left		visual	The `border-left` property setting is not applied to the parts of a form that are defined by <input>, <select>, <option> and <textarea> elements
	'border-left-width' || 'border-left-style' || 'border-left-color		
	inherit		
border		visual	The `border` property setting is not applied to the parts of a form that are defined by <input>, <select>, <option> and <textarea> elements
	'border-width' || 'border-style' || '<color>		
	inherit		

Visual formatting model

Property	Value	Media group	Remarks
display		all	
	block		
	inherit		
	inline		
	inline-table		**Ver 3.4 only**
	list-item		**Ver 3.4 only**. Only lists without numbers and lists with marker images are supported if `display:list-item` is specified

Property	Value	Media group	Remarks
	none		
	table-caption		**Ver 3.4 only**
	table-cell		**Ver 3.4 only**
	table-column		**Ver 3.4 only**
	table-column-group		**Ver 3.4 only**
	table-footer-group		**Ver 3.4 only**
	table-header-group		**Ver 3.4 only**
	table-row		**Ver 3.4 only**
	table-row-group		**Ver 3.4 only**
position		visual	
	static		
	relative		**Ver 3.4 only**. Only setting of `position:relative;` for block element is supported
	absolute		Unavailable HTML element: <hr>,<td>, <body>
	fixed		Unavailable HTML element: <hr>,<td>, <body>
	inherit		
top		visual	Has no effect
	<length>		
	<percentage>		
	auto		
	inherit		
right		visual	Has no effect
	<length>		
	<percentage>		
	auto		
	inherit		
bottom		visual	Has no effect
	<length>		

Property	Value	Media group	Remarks
	<percentage>		
	auto		
	inherit		
left		visual	Has no effect
	<length>		
	<percentage>		
	auto		
	inherit		
float		visual	Unavailable HTML element: <marquee>
	left		
	right		
	none		
	inherit		
clear		visual	
	none		
	left		
	right		
	both		
	inherit		
z-index		visual	The `z-index` property setting is not applied to the parts of a form that are defined by <input>, <select>, <button>, <textarea>, <iframe>, <embed> and <object> elements
	auto		
	<integer>		
	inherit		
direction			
	ltr		
	rtl		
	inherit		
unicode-bidi			Not supported

Visual formatting model details

Property	Value	Media group	Remarks
width		visual	
	<length>		
	<percentage>		
	auto		
	inherit		
min-width		visual	The `min-width` property setting is not applied to the following items: • float blocks, inline blocks • table cells
	<length>		
	<percentage>		
	inherit		
max-width		visual	The `max-width` property setting is not applied to the following items: • float blocks, inline blocks • table cells
	<length>		
	<percentage>		
	none		
	inherit		
height		visual	
	<length>		
	<percentage>		
	auto		
	inherit		
min-height			Not supported
max-height			Not supported
line-height		visual	
	normal		
	number		
	<length>		
	<percentage>		
	inherit		

Property	Value	Media group	Remarks
vertical-align		visual	
	baseline		
	middle		
	sub		
	super		
	text-top		
	text-bottom		
	<percentage>		
	<length>		
	top		
	bottom		
	inherit		

Visual effects

Property	Value	Media group	Remarks
overflow		visual	The `overflow` property setting is not applied to the parts of a form that are defined by <input>, <select>, <button>, <textarea>, <iframe>, <embed> and <object> elements
	visible		
	hidden		
	inherit		
clip		visual	
	auto		
	<shape>		**Ver 3.4 only**. Only specification by `rect(<top>, <right>, <bottom>, <left>)` is supported
	inherit		**Ver 3.4 only**
visibility		visual	
	visible		
	hidden		
	inherit		

Generated content, automatic numbering, and lists

Property	Value	Media group	Remarks
content			**Ver 3.4 only**. Only <string> is supported
quotes			Not supported
counter-reset			Not supported
counter-increment			Not supported
marker-offset			Not supported
list-style-type		visual	
	disc		
	circle		
	square		
	decimal		
	decimal-leading-zero		
	lower-roman		
	upper-roman		
	lower-alpha		
	lower-latin		
	upper-alpha		
	upper-latin		
	none		The marker is not displayed
	inherit		
list-style-image		visual	
	<uri>		
	none		
	inherit		
list-style-position		visual	
	inside		
	outside		
	inherit		
list-style		visual	

Property	Value	Media group	Remarks
	'list-style-type' || 'list-style-position' || 'list-style-image		
	inherit		

Paged media – not supported

Colors and Backgrounds

Property	Value	Media group	Remarks
color		visual	The `color` property setting is not applied to table columns or column groups defined by the <col> element or the <colgroup> element
	<color>		
	inherit		
background-color		visual	The effect of the `background-color` property setting on form parts defined by the <input>, <select>, <option> and <textarea> elements depends on the implementation of the window system
	<color>		
	transparent		
	inherit		
background-image		visual	The `background-image` property setting is not applied to the parts of a form that are defined by <input>, <select>, <option>, <textarea>, <img>, <iframe>, <embed> and <object> elements
	<uri>		
	none		
	inherit		
background-repeat		visual	The `background-repeat` property setting is not applied the table parts defined by the <thead> element
	repeat		
	repeat-x		

Property	Value	Media group	Remarks
	repeat-y		
	no-repeat		
	inherit		
background-attachment		visual	Available HTML elements: <html>, <body>
	scroll		
	fixed		
	inherit		
background-position		visual	
	<percentage>{1,2}		
	<length>{1,2}		
	[top | center | bottom] || [left | center | right]		
	inherit		
background		visual	The `background` property setting is not applied to the padding of the parts of a form that are defined by <input>, <select>, <option> and <textarea> elements when padding is specified
	'background-color' || 'background-image' || 'background-repeat' || 'background-attachment' || 'background-position		
	inherit		

Fonts

Property	Value	Media group	Remarks
font-family			Only one font in Sony Ericsson phones
	[[<family-name> | <generic-family>],]* [<family-name> | <generic-family>]	visual	
	inherit		
font-style		visual	
	normal		
	inherit		
font-variant		visual	
	normal		
	inherit		
font-weight		visual	
	normal		
	bold		
	100		Values 100 - 500 mapped to normal
	200		
	300		
	400		
	500		
	600		Values 600 - 900 mapped to bold
	700		
	800		
	900		
	inherit		
font-stretch			Not supported
font-size			Only 3 font sizes are supported in **W350, W380, Z310** and **Z555**; small, medium and large
	<absolute-size>	visual	
	<relative-size>		
	<length>		

Property	Value	Media group	Remarks
	<percentage>		
	inherit		
font-size-adjust			Not supported
font			
	['font-style' || 'font-variant' || 'font-weight']?'font-size' [/ 'line-height']? 'font-family	visual	
	caption		
	inherit		

Text

Property	Value	Media group	Remarks
text-indent		visual	The `text-indent` property setting is not applied to the table parts defined by the <table>, <caption>, <thead> and <td> elements
	<length>		
	<percentage>		
	inherit		
text-align		visual	
	left		
	right		
	center		
	inherit		
text-decoration		visual	
	underline || overline || line-through || blink		
	none		
	inherit		
text-shadow			Not supported

Property	Value	Media group	Remarks
letter-spacing			Not supported
word-spacing			Not supported
text-transform			Not supported
white-space			
	nowrap		**Ver 3.4 only**
	inherit		**Ver 3.4 only**

Tables

Property	Value	Media group	Remarks
caption-side		visual	
	top		
	bottom		
	inherit		
table-layout			**Ver 3.4 only**
	fixed		**Ver 3.4 only**. When the table-layout property and `width:auto` are set, display is always the same as when `table-layout:auto` is set. `display:inline-table` is not supported. The property value is applied to the <table> HTML element.
border-collapse		visual	
	collapse		
	separate		
	inherit		
border-spacing		visual	
	<length> <length>?		
empty-cells		visual	
	show		**Ver 3.4 only**
	hide		

Property	Value	Media group	Remarks
	inherit		
speak-header			aural media is not supported

User Interface (Interactive)

Property	Value	Media group	Remarks
cursor		visual	**Only** supported in phones with mouse cursor support.
	default		
	hand		
	pointer		
	progress		
outline			Not supported
outline-width			Not supported
outline-style			Not supported
outline-color			Not supported

Aural style sheets – not supported

WAP CSS Extension: Marquee

Property	Value	Media group	Remarks
display	-wap-marquee	visual	
-wap-marquee-style		visual	
-wap-marquee-loop		visual	
-wap-marquee-dir		visual	
-wap-marquee-speed		visual	

WAP CSS Extension: Access Keys

Property	Value	Media group	Remarks
-wap-accesskey		interactive	Fallbacks, Multiple Assignments and KeyCombination are not supported. Only ASCII code characters are supported, and values are ignored as disabled if two or more characters are specified.

WAP CSS Extension: Input

Property	Value	Media group	Remarks
-wap-input-format		interactive	
-wap-input-required		interactive	When the form is submitted, validity of entered data is checked

Scripting

This section contains lists of ECMAScript elements supported by the NF3 browser implementation in Sony Ericsson mobile phones. For details, please refer to the Standard ECMA-262 ECMAScript Language Specification found at http://www.ecma-international.org/publications/standards/Ecma-262.htm and the ECMAScript Mobile Profile specifications at http://www.openmobilealliance.org/.

The script elements are divided into the following subsections:

- Object properties and methods
- Statements
- Operators
- Types and literals.

Object properties and methods

Note: Only supported properties and methods are listed in the tables.

Global

	Property/method	Remarks
Value		
	`NaN`	
	`Infinity`	
	`undefined`	
	`version`	Processed only when ECMAScript Mobile Profile is enabled
Function		
	`eval(x)`	
	`parseInt(string [, radix])`	
	`parseFloat(string)`	
	`isNaN(number)`	
	`isFinite(number)`	
	`toString()`	JavaScript: String()
URI handling functions		
	`decodeURI(encodedURI)`	
	`decodeURIComponent(encodedURIComponent)`	

Property/method	Remarks
`encodeURI(uri)`	
`encodeURIComponent(uriComponent)`	
`escape(string)`	
`unescape(string)`	

Objects

Property/method	Remarks
Function call	
`Object([value])`	
Constructor	
`new Object([value])`	
Constructor object	
`length`	
`prototype`	
`version`	Processed only when ECMAScript Mobile Profile is enabled
Prototype object	
`constructor`	
`toString()`	
`toLocaleString()`	
`valueOf()`	
`hasOwnProperty(V)`	
`isPrototypeOf(V)`	
`propertyIsEnumerable(V)`	

Function

Property/method	Remarks
Function call	
`Function([p1, p2, ... , pN,] body)`	
Constructor	

Property/method	Remarks
`new Function([p1, p2, ... , pN,] body)`	
Constructor object	
`length`	
`prototype`	
`version`	Processed only when ECMAScript Mobile Profile is enabled
Instance	
`length`	
`prototype`	
Prototype object	
`constructor`	
`toString()`	
`apply(thisArg [, argArray])`	
`call(thisArg [, arg1, arg2, ... , argN])`	

Arguments

Property/method	Remarks
`callee`	
`length`	

Array

Property/method	Remarks
Function call	
`Array([item1, item2, ... , itemN])`	
Constructor	
`new Array([item0, item1, ... , itemN])`	
`new Array(len)`	
Constructor object	
`length`	
`prototype`	

Property/method	Remarks
`version`	Processed only when ECMAScript Mobile Profile is enabled
Prototype object	
`constructor`	
`toString()`	
`toLocaleString()`	
`concat([item1, item2, ... , itemN])`	
`join(separator)`	
`pop()`	
`push([item1, item2, ... , itemN])`	
`reverse()`	
`shift()`	
`slice(start [, end])`	
`sort(comparefn)`	
`splice(start, deleteCount [, item1, item2, ... , itemN])`	
`unshift([item1, item2, ... , itemN])`	
Instance	
`length`	

String

Property/method	Remarks
Function call	
`String([value])`	
Constructor	
new String([value])	
Constructor object	
`length`	
`prototype`	

	Property/method	Remarks
	`fromCharCode([code0, code1, ... , codeN])`	
	`version`	Processed only when ECMAScript Mobile Profile is enabled
Prototype object		
	`constructor`	
	`toString()`	
	`valueOf()`	
	`charAt(pos)`	
	`charCodeAt(pos)`	
	`concat([item1, item2, ... , itemN])`	
	`indexOf(pattern [, pos])`	
	`lastIndexOf(pattern [, pos])`	
	`localeCompare(that)`	
	`match(regexp)`	
	`replace(searchValue, replaceValue)`	
	`search(regexp)`	
	`slice(start [, end])`	
	`split(separator [, lim])`	
	`substring(start, end)`	
	`substr(start [, length])`	
	`toLowerCase()`	
	`toLocaleLowerCase()`	
	`toUpperCase()`	
	`toLocaleUpperCase()`	
	`anchor(string)`	
	`link(string)`	
	`fontcolor(string)`	
	`fontsize(string)`	
	`big()`	
	`blink()`	

	Property/method	Remarks
	`bold()`	
	`fixed()`	
	`italics()`	
	`small()`	
	`strike()`	
	`sub()`	
	`sup()`	
Instance		
	`string.length`	

Boolean

	Property/method	Remarks
	Function call	
	`Boolean(value)`	
Constructor		
	`new Boolean(value)`	
Constructor object		
	`length`	
	`prototype`	
	`version`	Processed only when ECMAScript Mobile Profile is enabled
Prototype object)		
	`constructor`	
	`toString()`	
	`valueOf()`	

Number

	Property/method	Remarks
Function call		
	`Number([value])`	
Constructor		
	`new Number([value])`	
Constructor object		

	Property/method	Remarks
	length	
	prototype	
	MAX_VALUE	
	MIN_VALUE	
	NaN	
	NEGATIVE_INFINITY	
	POSITIVE_INFINITY	
	version	Processed only when ECMAScript Mobile Profile is enabled
Prototype Object		
	constructor	
	toString([radix])	
	toLocaleString()	
	valueOf()	
	toFixed(fractionDigits)	
	toExponential(fractionDigits)	
	toPrecision(precision)	

Math

	Property/method	Remarks
Values		
	E	
	LN10	
	LN2	
	LOG2E	
	LOG10E	
	PI	
	SQRT1_2	
	SQRT2	
	version	Processed only when ECMAScript Mobile Profile is enabled
Functions		

Property/method	Remarks
`abs(x)`	
`acos(x)`	
`asin(x)`	
`atan(x)`	
`atan2(y, x)`	
`ceil(x)`	
`cos(x)`	
`exp(x)`	
`floor(x)`	
`log(x)`	
`max([value1, value2, ... , valueN])`	
`min([value1, value2, ... , valueN])`	
`pow(x, y)`	
`random()`	
`round(x)`	
`sin(x)`	
`sqrt(x)`	
`tan(x)`	

Date

Property/method	Remarks
Function call	
`Date ([year, month, date, hours, minutes, seconds, ms])`	
Constructors	
`new Date(year, month [, date, hours, minutes, seconds, ms])`	
`new Date()`	
`new Date(milliseconds)`	
`new Date(dateString)`	

Property/method	Remarks
Constructor object	
`length`	
`prototype`	
`parse(string)`	
`UTC(year, month [, date, hours, minutes, seconds, ms])`	
`version`	Processed only when ECMAScript Mobile Profile is enabled
Prototype object	
`constructor`	
`toString()`	
`toDateString()`	
`toTimeString()`	
`toLocaleString()`	
`toLocaleDateString()`	
`toLocaleTimeString()`	
`valueOf()`	
`getTime()`	
`getFullYear()`	
`getUTCFullYear()`	
`getMonth()`	
`getUTCMonth()`	
`getDate()`	
`getUTCDate()`	
`getDay()`	
`getUTCDay()`	
`getHours()`	
`getUTCHours()`	
`getMinutes()`	
`getUTCMinutes()`	
`getSeconds()`	
`getUTCSeconds()`	

Property/method	Remarks
`getMilliseconds()`	
`getUTCMilliseconds()`	
`getTimezoneOffset()`	
`setTime(time)`	
`setMilliseconds(ms)`	
`setUTCMilliseconds(ms)`	
`setSeconds(sec [, ms])`	
`setUTCSeconds(sec [, ms])`	
`setMinutes(minute [, sec, ms])`	
`setUTCMinutes(minute [, sec, ms])`	
`setHours(hour [, minute, sec, ms])`	
`setUTCHours(hour [, minute, sec, ms])`	
`setDate(date)`	
`setUTCDate(date)`	
`setMonth(month [, date])`	
`setUTCMonth(month [, date])`	
`setFullYear(year [, month, date])`	
`setUTCFullYear(year [, month, date])`	
`toUTCString()`	
`toGMTString()`	
`getYear()`	
`setYear(year)`	

RegExp

Property/method	Remarks
Function call	
`RegExp(pattern [, flags])`	
Constructor	

Property/method	Remarks
new RegExp(pattern [, flags])	
Constructor object	
`length`	
`prototype`	
`version`	Processed only when ECMAScript Mobile Profile is enabled
Prototype object	
`constructor`	
`exec([string])`	
`test([string])`	
`toString()`	
`valueOf()`	
Instances	
`source`	
`global`	
`ignoreCase`	
`multiline`	
`lastIndex`	

Error

Property/method	Remarks
Function	
`Error(message)`	
Constructor	
`new Error(message)`	
Constructor object	
`length`	
`prototype`	
`version`	Processed only when ECMAScript Mobile Profile is enabled
Prototype object	
`constructor`	

Property/method	Remarks
`name`	
`message`	
`toString()`	
`code`	Processed only when ECMAScript Mobile Profile is enabled. Memory error does not occur

EvalError, RangeError, ReferenceError, SyntaxError, TypeError, URIError

These objects have the same properties and methods as the Error object above.

Statements

Variable statement

Syntax	Remarks
`var varname [= value] [..., varname [= value]]`	

Empty statement

Syntax	Remarks
`;`	

if statement

Syntax	Remarks

`if ( condition )`
`statements1`
`}`
`[else {`
`statements2`
`}]`	

Iteration Statements

Syntax	Remarks
`do statements` `while ( condition );`	The browser has a predefined limit for iterations to avoid infinite loops. If the limit is reached, script execution stops

Syntax	Remarks

`while ( condition ) {`
`statements`
`}`
	The browser has a predefined limit for iterations to avoid infinite loops. If the limit is reached, script execution stops

`for ([initial-expression]; [condition]; [increment-expression]) {`
`statements`
`}`
	The browser has a predefined limit for iterations to avoid infinite loops. If the limit is reached, script execution stops

`for ( variable in object ) {`
`statements`
`}`
	The browser has a predefined limit for iterations to avoid infinite loops. If the limit is reached, script execution stops

continue statement

Syntax	Remarks
`continue [ label ]`	

break statement

Syntax	Remarks
`break [ label ]`	

return statement

Syntax	Remarks
`return expression`	

with statement

Syntax	Remarks

`with ( object ) {`
`statements`
`}`	

label statement

Syntax	Remarks
`label :` `statements`	

switch statement

Syntax	Remarks

```
switch ( expression ) {
case label :
  statements;
  break;
case label :
  statements;
  break;
...
default : statements;
}
```

throw statement

Syntax	Remarks

```
throw expression
```

try statement

Syntax	Remarks

```
try {
statements
}
[catch ( catchID ) {
statements
}]
[finally {
statements
}]
```

function statement

Syntax	Remarks

```
function name([ param][,param] [...,
param]) {
statements
}
```

Comments

Syntax	Remarks

```
// comment text
/* multiple line comment text */
```

Operators

Arithmetic operators

Operators

```
x ++
++ x
+ x
x --
-- x
- x
x + y
x - y
x * y
x / y
x % y
```

Comparison operators

Operators

```
x == y
x != y
x === y
x !== y
x > y
x >= y
x < y
x <= y
```

Bitwise operators

Operators

```
x & y
x | y
x ^ y
~ x
x << y
x >> y
x >>> y
```

Logical operators

Operators

```
expr1 && expr2
expr1 || expr2
! expr
```

Assignment operators

Operators

```
x = y
x += y
x -= y
x *= y
x /= y
x %= y
x <<= y
x >>= y
x >>>= y
x &= y
x ^= y
x |= y
```

String operators

Operators

```
string + string
string += string
```

Conditional operator

Operators

```
condition ? expr1 : expr2
```

Comma operator

Operators

```
expr1, expr2
```

delete operators

Operators

```
delete objectName
delete objectName. property
delete objectName[index]
delete property
```

new operator

Operators

```
objectName = new objectType ( param1 [, param2] ...[, paramN])
```

typeof operators

Operators

```
typeof operand
typeof (operand)
```

void operators

Operators

```
void (expression)
void expression
```

instanceof operator

Operators

```
objectName instanceof objectType
```

in operator

Operators

```
propNameOrNumber in objectName
```

this keyword

Operators

```
this[. propertyName]
```

Others

Operators

```
[ ]
( )
.
```

Types and literals

Types

```
Undefined
Null
Boolean
Number
String
Object
```

Literals

```
Boolean values
Numbers
Strings
Array
Object
null
```

URI schemes

URI (Uniform Resource Identifier) schemes are used to identify resources, such as Web pages, on the Internet. The most common URI scheme is “http”, which is used to identify a resource (for example a Web page or an image) on a Web server.

This section defines the URI schemes that can be used in the Web browser, in addition to the common “http” and “https” schemes. The NF3 browser in Sony Ericsson phones supports the tel, wtai, sms, smsto, mms, mmsto, mailto, http, https and rtsp URI scemes.

Telephone numbers: the “tel” URI scheme

The “tel” URI scheme specifies a phone number. When the scheme is invoked, the browser asks the user whether to place a voice call, a video call (if supported), send a text messge (SMS), or add the phone number to the phonebook.

On many Web pages the “mailto” scheme is used to let visitors contact the page author or site administrator. When the page is viewed from a phone, placing a call may be more appropriate.

In V4 you can also place video calls using the “tel” scheme.

Example
```
<p><a href="tel:+46905123456">Call me</a></p>
```

A sequence of DTMF tones can be added to the phone number by using the “postd” parameter. When the call is established, the user is requested to send the DTMF tones. Use this to create links to voice services controlled by DTMF tones: helpdesk, voicemail, and so on. When the scheme is invoked with the “postd” parameter, the user is prompted whether to place a voice call or add the number to the phonebook, that is, it is not possible to send a text message.

Example
```
<p>
<a href="tel:+46905123456;postd=2922%23">
Customer service</a>
<a href="tel:+46905133133;postd=9%23">
Voice mail</a>
<a href="tel:+46905166166;postd=92%23">
Helpdesk</a>
</p>
```

See also “URLs for Telephone Calls”, RFC2806, http://www.ietf.org/rfc/rfc2806.txt

Note: The “#” character has a special meaning in URLs. In a post dial string a “#” character must therefore be replaced with the escape sequence “%23” as in the example above.

SMS message 1: the “smsto” URI scheme

The “smsto” URI scheme specifies a phone number to a phone that can receive text messages (all GSM phones). When the user selects the link the text message editor opens with the specified phone number as the recipient.

Example
```
<a href="smsto:+46905123456">Send me an SMS!</a>
```

In V4, the “body” parameter can be used to pre-fill the message.

Example
```
<a href="smsto:+46905123456?body=Subscribe">smsto:body</a>
```

SMS message 2: the “sms” URI scheme

The "sms" URI scheme is the same as the "smsto" scheme. Only the name of the scheme is different.

Example
```
<a href="sms:+46905123456">Send me an SMS!</a>

<a href="sms:+46905123456?body=subscribe">Subscribe</a>
```

MMS (picture message): the “mms” and “mmsto” URI schemes

The "mms" and “mmsto” URI schemes behave in a similar way as the “sms” scheme. When the user clicks the link, the MMS editor in the phone opens and the phone number in the link is set as recipient of the message.

Example
```
<a href="mms:+46905123456"></a>

<a href="mmsto:+46905123456"></a>
```

E-mails: the “mailto” URI scheme

The “mailto” URI scheme specifies an email address, and is commonly used on the Web. When the user selects the link, the email editor opens with the specified address as the recipient. The “to”, “cc”, “subject”, and “body” parameters can be used to pre-populate these fields.

Example
```
<p>
<a href="mailto:me@example.com">Send me an e-mail!</a>


<a href="mailto:mail-
list@example.com?cc=name@example.com&subject=Check this
Out!&Body=Here is a lot of text for the body field of the
message.">
Mail me</a>
</p>
```

See also
“The 'mailto' scheme”, http://www.ietf.org/rfc/rfc2368.txt

Telephony: the “wtai” URI scheme

The “wtai” scheme specifies a function and a number of parameters depending on the function.

The function “//wp/mc” establishes a phone call to the specified number.

The function “//wp/ap” lets the user add the specified number, and optionally a name, to the phonebook.

The function "//wp/sd" lets the user send DTMF tones, after a call has been established. The user must first set up a phone call with, for example the "//wp/mc" function, before sending DTMF tones.

The different functions of the "wtai" scheme can be used to create a Web page that first lets the user establish a phone call (with "wtai://wp/mc"), and then, when the call is established, select links in the page to invoke different services (with "wtai://wp/sd"). Thus, the user interacts with both the Web browser and the telephony call at the same time, i.e. a simple kind of "multi-modal" service.

Example
```
<p>
<a href="wtai://wp/mc;+46905123456">
Call Helpdesk</a>
<a href="wtai://wp/ap;+46905123456;Helpdesk">
Add Helpdesk to phonebook</a>
</p>
```

See also OMA WTAI Public http://www.openmobilealliance.com/

Note: The “wtai” scheme does essentially the same things as the “tel” scheme, but uses a different syntax.

Streaming: the “rtsp” URI scheme

The "rtsp" URI scheme specifies a media stream. When the user selects the URI, the Web browser starts the media player, which sets up a media stream to the streaming server.

Example `<a href="rtsp://example.com/stream.3gp">Start media stream</a>`

A media stream can also be started by an "http" URL to an SDP (Session Description Protocol) file on a standard Web server.

Example `<a href="http://example.com/stream.3gp.sdp">Start media stream</a>`

Sony Ericsson have developed a special plugin allowing streaming video to be played in the browser. For more information about the features of this plugin and how to invoke it, see “Streaming video in an embedded window” on page 91.

Java MIDlet URI schemes

Java MIDlets can register themselves for other URI schemes than those above.

When the browser detects any other URI scheme than those above, it automatically calls the JSR-211 API in the phone, and if the scheme has been registered by a Java MIDlet, this MIDlet is executed.

Web feeds (RSS feeds)

Later Sony Ericsson phones include a Web feed functionality, based on the RSS 2.0 specification. Web feeds are handled by a separate application, which seemlessly interacts with the Web browser in handling the creation of Web feeds. For information about which phones support Web feeds, see “Products” on page 3.

For detailed specifications of the Web feed implementation in Sony Ericsson phones, see Developers guidelines – RSS, found at http://developer.sonyericsson.com/getDocument.do?docId=99515

Server-side development

When the browser sends a request to a Web server, it reveals information about itself, such as the phone model and software version. The server can use the information to adapt the site to the capabilities of the phone.

Identifying the browser

Information about the browser can be primarily found in these two places:

- The standard HTTP User-Agent header field, which is a text string that contains the phone model and the release number. If the phone supports Java, information about the Java version is also included.
- The Profile header field, which contains a URL that points to a UAPROF profile located at the Sony Ericsson Web site.

Of the above, the HTTP User-Agent header field is a de facto Web standard and sent by all Web browsers. For exampe, IE6 sends the User-Agent header field “Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)”. The UAPROF profile is sent by most mobile phones that support the OMA standard. The profile is an RDF/XML document that contains detailed information about the capabilities of the phone, such as supported character encoding, MIME types and screen size.

The following table lists examples of User-Agent header fields and the corresponding UAPROF profile URLs for a selection of Sony Ericsson phone models:

Phone	User-Agent header	UAPROF profile URL
T610	`SonyEricssonT610/R501 Profile/MIDP-1.0 Configuration/CLDC-1.0`	http://wap.sonyericsson.com/UAprof/T610R501.xml
K700i	`SonyEricssonK700i/R2AG SEMC-Browser/4.0.3 Profile/MIDP-2.0 Configuration/CLDC-1.1`	http://wap.sonyericsson.com/UAprof/K700iR201.xml
W550i	`SonyEricssonW550i/R1A Browser/NetFront/3.3 Profile/MIDP-2.0 Configuration/CLDC-1.1`	http://wap.sonyericsson.com/UAprof/W550iR101.xml

User Agent profile URLs for all Sony Ericsson phones are published in a White paper on Developer World: http://developer.sonyericsson.com/getDocument.do?docId=65048

The User-Agent header field

The User-Agent header field consists of a list of tokens. A token is a short text string, which may, for example, indicate the browser version and the phone version.

The following sections describe the tokens for the phone model/software version and the browser version. The header field may contain other, additional tokens, for example, to indicate support for Java.

Phone model and software version

The phone model and software version are indicated in a token with the following format:

"SonyEricsson" model "/" release

model is the phone model, for example "W550i".

`release` indicates the software version of the phone, for example "R1A" or "R1B".

Browser version

The version of the browser is indicated in the "Browser" token.

```
User-Agent: Browser/NetFront/3.3
```

```
User-Agent: Browser/NetFront/3.4
```

User agent profile (UAPROF)

In the request to the server, the browser includes a reference to a user agent profile (UAPROF). In the profile, which is an RDF/XML document located on the Sony Ericsson Web site, the server can find more detailed information about the browser than that available in the HTTP request headers.

The UAPROF is part of the overall Composite Capabilities/Preferences Profile (CC/PP) framework, which is defined by the W3C at http://www.w3.org/Mobile/CCPP/.

Profiles for Sony Ericsson browsers are located in the http://wap.sonyericsson.com/UAprof/ directory. Profiles for other phones are linked from http://w3development.de/rdf/uaprof_repository/.

The name of a Sony Ericsson profile for a particular browser is composed of the phone model name, the release number, and the ".xml" file-extension. The name has the following format:

```
model release ".xml"
```

Example: `W550iR101.xml`

When the request is sent to the server, it passes the WAP gateway. The gateway translates the request from the browser into a standard HTTP request. In this translation, the reference to the profile is translated into a set of HTTP header fields, according to the CCPP Exchange protocol (see http://www.w3.org/TR/NOTE-CCPPexchange for details) and the HTTP Extension framework (see RFC2774 for details).

To find the profile reference in the HTTP request when it arrives to the server, you need to do the following:

- Look up all Opt header fields in the HTTP request. The Opt header field declares an HTTP extension: a unique URL and a namespace number.

- Look up the Opt header which contains the URL <http://www.w3.org/1999/06/24-CCPPexchange>. This is the CCPP extension identifier. The “ns” parameter of the header is the namespace for CCPP.
  `Opt: "http://www.w3.org/1999/06/24-CCPPexchange" ; ns=56`

- Look up all profile header fields that start with the namespace number from the CCPP extension identifier. In the above example, the identifier is “56”, so the profile header field looks like this:

```
56-Profile: http://wap.sonyericsson.com/UAprof/W550iR101.xml
```

The above steps are executed on the server by a script program or some other server program. Here is an example of a complete HTTP request with a UAPROF reference:

```
GET / HTTP/1.1
Host: www.example.com
...
Opt: "http://www.w3.org/1999/06/24-CCPPexchange" ; ns=80
80-Profile: http://wap.sonyericsson.com/UAprof/T68R501.xml
```

Cache

The browser temporarily saves caches, images and Web pages according to the HTTP standard, http://www.ietf.org/rfc/rfc2616.txt, which all Web browsers conform to.

The cache is controlled by the HTTP Cache-control header field. The most common cache control directive is to turn caching off for certain Web pages.

```
Cache-control: no-cache
```

The “no-cache” directive prevents the browser from putting the page into the cache, so the page is always requested from the server.

Note: Always turn caching off on Web pages that contain content that changes often, such as news, otherwise the user may look at old Web pages - and old news.

Download

Many Web sites let the user download contents such as ringtones and themes. Compared with navigating from one Web page to another, downloading is different in many ways. The content may be too large to store in the phone, it may be in a format that is not supported, or it may cost money.

Download methods

There are primarily two ways by which a Web page can refer to downloadable contents:

- **Direct link**: A standard hyperlink, the <a> element, refers to the content. When the user selects the link, the content is downloaded.
- **OMA Download**: Before the actual content is downloaded, a small XML document, the download descriptor, is downloaded with information about the content, such as its name, size and vendor. The descriptor is used by the browser to determine whether the content can be supported on the phone. It may also be used to report the outcome of the download back to the content provider.

The user interface is similar in both cases. A prompt requests the user to download the content. A progress bar indicates the progress of the download. When the content is received its type determines how it is handled. An image is placed in the image folder, a ringtone is placed among the ringtones, and so on.

The direct link and download descriptor methods are described in the following sections.

Maximum content size

The section “Web pages and other content” below describes how the browser distinguishes Web pages from other content types. The browser uses two different methods for download, depending on the type of content. Developers who want to make sure that the browser is able to download large content, must make sure that the browser does not treat it as a Web page.

The protocol runs over a TCP/IP connection. The only limiting factor is how much memory is available when the download takes place. The amount of memory varies between phone models and also depends on whether a Memory Stick™ is used.

Direct link

The simplest way of creating a Web page with downloadable content is to use a direct link, the <a> element.

```
<a href="image.gif">Download this image</a>
```

When the user selects the link, the browser prompts the user whether to download the content, view information about the content, or cancel the operation.

In this case, the only information the browser has about the content is the URL from the <a> element. The size is not available until the browser requests the content from the server.

The HEAD method

Note: This section assumes the reader has basic knowledge of HTTP, the standard transfer protocol on the Web. The specification is available at http://www.ietf.org/rfc/rfc2616.txt.

For all content except Web pages, the browser sends a HEAD method to the server before downloading the actual content. The HEAD method is used to retrieve information about the content, such as its size and media type. The information is used to determine whether the phone is able to support the content.

Use of the HEAD method increases accuracy of information about the content that is presented to the user. Before download of content that is not supported by the phone, the user is prompted that the file type is not supported, but is given the choice to download anyway.

OMA download

OMA download enables advanced multimedia download services.

Note: The OMA download 1.0 standard is available from the OMA Web site, http://www.openmobilealliance.org/.

Here are some examples of when OMA download can be used when it is important that:

- The user is able to view information about the content, and is given a chance to choose whether to download it or not, before the file is downloaded
- The phone checks that the content is not too large, based on the currently available memory in the phone, and that the MIME media type is supported. If the media type is not supported, the user is informed of this and is given the choice whether to download or not
- The content provider is given an indication of the outcome of the download.

The browser supports the OMA download protocol, which provides a solution to the above requirements. It does so by introducing a download descriptor, a small XML document which describes the file and is downloaded before the actual file is requested.

Here is an example of a simple descriptor for a GIF image:

```
<?xml version="1.0" encoding="UTF-8" ?>
<media>
  <name>Homer</name>
  <description>Homer speaking in mobile phone</description>
  <type>image/gif</type>
  <size>2000</size>
  <objectURI>
```

```
    http://www.example.com/images/
  </objectURI>
  <installNotifyURI>
    http://www.example.com/cgi-bin/install.pl
  </installNotifyURI>
</media>
```

When the browser receives the descriptor, it presents the <name>, <description> and <size> elements to the user. It also checks the size and the MIME media type to make sure the phone will be able to handle the file, should the user choose to download it. The file is located at the URL in the <objectURI> element.

After the download of the object pointed out in the descriptor has taken place, the browser sends (using HTTP POST method) a status report to the URL in the <installNotifyURI> element. At that URL the content provider can run an application that creates some kind of log file to keep track of all downloads, generates a charging record, or sends a notification to someone that a download has taken place. If sending the report fails, the content will be removed.

A descriptor must contain the following elements: <type>, <size>, and <objectURI>.The elements must be inside the <media> root element. The following sections describe all the elements.

<DDVersion>

This element specifies the version of the OMA standard to which the descriptor conforms. The browser supports version “1.0”, the only version that exists today. If the element is not present, version “1.0” is the default.

Value 1.0

<description>

This element contains a textual description of the file. It is important that the text is meaningful to the user. For example, the text “a GIF image” will not help the user to decide whether to download the file or not. “Homer speaking on a mobile phone” gives the user more information.

Value A text string

<installNotifyURI>

This element contains the address of the server application, to which the browser posts a status report of the outcome of the download.

Here are the possible status reports and when they are sent by the browser:

Status report	**Sent when**
`900 Success`	Success
`901 Insufficient memory`	Not enough memory
`902 User cancelled`	The user cancels the download
`903 Loss of service`	Network is down, and so on
`905 Attribute mismatch`	The actual file does not match its description, for example larger of a different MIME media type than what was in the download descriptor

`906 Invalid descriptor`	XML syntax error or unexpected element content (for example a URL instead of an integer)
`952 Device aborted`	Some error occurred in the phone, which was not one of the above
`953 Non-acceptable content`	The MIME media type is not supported
`954 Loader error`	The file was unavailable at the URL

The complete list of all status messages is available in the OMA Download specification at http://www.openmobilealliance.org/.

The status report is provided in the body of the POST message. Here is an example of a report for a successful download:

```
POST /cgi-bin/installnotify.pl
Host: www.example.com
Content-length: 11

900 Success
```

When the status report has been received, the server should respond with a simple HTTP status code (for example “200 Success”), otherwise the browser will display an error message to the user saying that the session was aborted and the object has been removed from the phone.

Value: A URL

<name>

This element specifies the name of the file. The value is presented to the user.

Value A text string

<objectURI>

This element is the URL location of the file.

Value A URL

<size>

The content of this element is presented to the user as the size of the file, in bytes.

If the file is too large for the browser to download or save, an error message is displayed and the file is not downloaded. It is better to indicate too small a number, than one that is too large.

Value An integer

<type>

An error message is displayed if the browser does not support the MIME media type specified in this element, and the file has not been downloaded. In the case of DRM-wrapped objects the type specified may be either the object or DRM type.

Here are some examples of common MIME media types:

image/gif	GIF image	image/vnd.wap.wbmp	WBMP image
image/jpeg	JPEG image	audio/midi	MIDI ringtone

Value: A MIME media type

<vendor>

The content of this element is presented to the user as the file vendor, the person or organisation that created the file.

Value A text string

Appendix 1 – Phone features

This sections lists a few technical features for the phones mentioned in this document. More technical information about Sony Ericsson mobile phones can be found at Sony Ericsson Developer World, http://developer.sonyericsson.com/site/global/docstools/phonespecs/p_phonespecs.jsp.

Features

Feature	Phone series			
	W550, W600	**W900**	**W810**	**K530, K550, K610, K630, V640, W350, W380, W610, W660, W710, Z550, Z555, Z558, Z610, Z710**
Display size	176x220 px	240x320 px	176x220 px	176x220 px
Display colour depth	18-bit 262,144 colours	18-bit 262,144 colours	18-bit 262,144 colours	18-bit 262,144 colours
Max memory available	256 MB	470 MB	W810: 21 MB	W710, Z710: 10 MB W350, W380, Z555: 14 MB K530, W660: 16 MB Z558: 18 MB Z550: 20MB K630, V640: 32 MB K550, K610, W610, Z610: 64 MB
Memory Stick PRO Duo™ and Memory Stick Duo™ support	No	Yes	Yes	No
Memory Stick Micro™ (M2™) support	No	No	No	Yes
Cache size (cache deleted when the phone is turned off)	300 KB	300 KB	300 KB	300 KB

Feature	Phone series		
	C510, C702, C901, C902, C903, C905, G502, G705, K660, K770, K790, K800, K810, K850, S500, T650, T700, T707, W508, W580, W595, W705, W715, W760, W830, W850, W880, W890, W902, W910, W980, W995, Z750, Z770, Z780	**K310, K320, W200, Z310, Z530**	**K510, W300**
Display size	240x320 px	128x160 px	128x160 px
Display colour depth	18-bit 262,144 colours	16-bit 65,536 colours	18-bit 262,144 colours
Max memory available	W980: 8 GB C702, C902, C905: 160 MB C510, C903, G705, W508, W705, W715: 120 MB W995: 118 MB C901: 110 MB T707: 100 MB T650: 50 MB K850, W595, W760, W910: 40 MB Z780: 35 MB K660, W890, G502, Z750, Z770: 32 MB T700: 25 MB W902: 24 MB K770, K790, K800, K810, W830, W850, W880: 16 MB S500, W580: 12 MB	Z530: 28MB W200: 20MB K310, K320: 15MB Z310: 14MB	K510: 28MB W300: 20MB
Memory Stick PRO Duo™ and Memory Stick Duo™ support	W830, W850		
Memory Stick Micro™ (M2™) support	C510, C702, C901, C902, C903, C905, G502, G705, K660, K770, K790, K800, K810, K850, S500, T650, T700, T707, W508, W580, W595, W705, W715, W760, W880, W890, W902, W910, W995, Z750, Z770, Z780	W200, Z530	W300
SanDisk microSD™/Trans-flash™ support	K850		
Cache size (cache deleted when the phone is turned off)	300 Kb + 1MB on Memory Stick (when used)		

Appendix 2 – Examples

This appendix contains examples of specific applications for the NetFront browser in Sony Ericsson mobile phones.

Streaming video in an embedded window

Note: This functionality is supported only in the C702, C901, C902, C903, C905, G502, K530, K550, K630, K660, K770, K810, K850, S500, T650, T700, T707, V640, W595, W610, W660, W710, W715, W760, W830, W850, W880, W890, W902, W910, W980, W995, Z610, Z710, Z770 and Z780 series and in late K610, K790 and K800 phones. Early K610, K790 and K800 phones do not support the functionality, but may do so after updating the phone software to the latest available version.

To allow streaming video to be played in a window embedded in a web page, Sony Ericsson have developed a special player plugin for the browser. Control functions for the player, like PLAY, PAUSE and FULLSCREEN are handled via the phone softkeys. However, the web page with the embedded media window is still active, and may contain links to other streaming content, allowing the user to “change channel” by selecting a link to another streaming video on the web page.

The embedded player requires either a streaming account or an Internet account set in the phone to be able to play video. The player first tries to connect using a streaming account, and if no such account is set up, it tries using an available Internet account instead.

Starting the embedded player

A link to the video to be streamed with the plugin is defined as <embed> or <object>, for example:

```
<embed id="someID" width="176" height="132" src="rtsp://xxx.xxx.xxx.xxx/
someStreamingVideo.3gp" type="video/3gpp" autostart="true"></embed>
```

or

```
<object id="someID" width="176" height="132" data="rtsp://xxx.xxx.xxx.xxx/
someStreamingVideo.3gp" type="video/3gpp" autoplay="false"></object>
```

If no size is defined with the `width` and `height` attributes, the NF3 browser draws a default size window. The plugin window is clipped only by phone screen size limitations. If, for example, the plugin is contained in an iFrame smaller than the plugin window, it will not be clipped but will “shine through” the areas surrounding the iFrame.

Until a video is playing, the plugin window is blank (dark grey). The `Autoplay` and `Autostart` attributes are supported. When set to `True`, the video starts playing immediately when the plugin window becomes visible. The default value is `True`.

Only one plugin window can be active at a time. If the page contains more than one embedded player in the code, only the first one can play video, any nextcoming players are shown as “broken”, indicated by a very light gray (almost white) background colour.

Allowed `Type` attribute values:

- `video/3gp`
- `video/3gpp`

If errors occurs, for example, if no streaming or Internet account is set in the phone, the media server does not respond, or the video cannot be played due to an invalid proxy, error messages and dialogs are handled by the plugin.

After the plugin has been initiated on the page, the option list accessed via the Right Softkey (RSK) of the phone contains the selectable option Play if the video is not already playing. If the video is playing, Stop and Fullscreen are the selectable options in the list instead.

Changing channel

The user can still navigate on the webpage when a video is playing in the plugin window. If he chooses a link with the javascript `PlaySource(pURL)` the current video (if any) is stopped, and the video with the URL `pURL` is started in the player. Example:

```
document.getElementById("someID").PlaySource(pURL);
```

Stopping

If a stream is playing, the user can stop playing by choosing the Stop option from the option list accessed via the Right Softkey of the phone.

Playing interrupted by an incoming call or SMS

If the user receives an incoming call or SMS while a video is playing, the browser loses focus and the stream stops playing until the plugin regains playback focus.

Closing the player

When the user leaves the page with the embedded player, or closes the browser, the player is automatically destroyed and all associated data is lost.

Index

W

X