

---

# **Introduction to Creating Mobile Services for Series 60 Browser**

**Version 1.0**  
March 2003

S E R I E  
**60** - A T F O R M

## Table of Contents

<b>1.</b>	<b>Introduction.....</b>	<b>6</b>
1.1	Purpose and Scope.....	6
1.2	Series 60 Mobile Browser .....	6
1.3	References .....	6
<b>2.</b>	<b>Creating Mobile Services for Series 60 Browser .....</b>	<b>7</b>
2.1	Effective XHTML/WML Applications .....	7
2.1.1	Content .....	7
2.1.2	Design.....	7
2.2	Dynamic XHTML/WML Applications .....	7
2.2.1	Creating dynamic content.....	7
2.2.2	WAP Push .....	8
<b>3.</b>	<b>Development Tools.....</b>	<b>9</b>
3.1	SDKS and IDEs.....	9
3.1.1	Nokia Mobile Toolkit 3.1 .....	9
3.1.2	Series 60 Content Authoring SDK 1.0 for Symbian OS, Nokia edition .....	9
3.1.3	Ericsson WAP IDE 3.2.1.....	9
3.2	Image Tools.....	9
3.3	Other Helpful Applications.....	9
<b>4.</b>	<b>Writing A Simple XHTML MP Application .....</b>	<b>10</b>
4.1	Application Essentials .....	10
4.1.1	XHTML MP .....	10
4.1.2	Style sheets .....	10
4.2	Creating a Simple XHTML Page.....	11
4.2.1	Using Nokia Mobile Internet Toolkit 3.1.....	11
4.2.2	Adding a style sheet .....	13
<b>5.</b>	<b>Publishing an XHTML/WML Application .....</b>	<b>16</b>
5.1	How a Series 60 Browser Loads an Application .....	16
5.2	MIME Types .....	16
<b>6.</b>	<b>Summary .....</b>	<b>18</b>

## Version History

Version	Date	Comment
1.0	07/03/2003	Initial document release

## Legal Notice

Copyright © Nokia Corporation 2003. All rights reserved.

Reproduction, transfer, distribution or storage of part or all of the contents in this document in any form without the prior written permission of Nokia is prohibited.

Nokia and Nokia Connecting People are registered trademarks of Nokia Corporation. Other product and company names mentioned herein may be trademarks or tradenames of their respective owners.

Nokia operates a policy of continuous development. Nokia reserves the right to make changes and improvements to any of the products described in this document without prior notice.

Under no circumstances shall Nokia be responsible for any loss of data or income or any special, incidental, consequential or indirect damages howsoever caused.

The contents of this document are provided "as is." Except as required by applicable law, no warranties of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose, are made in relation to the accuracy, reliability or contents of this document. Nokia reserves the right to revise this document or withdraw it at any time without prior notice.

## Glossary

Term	Meaning
<b>CSS</b>	Cascading Style Sheet. A mechanism for adding style (e.g., fonts, colors, spacing) to Web documents.
<b>DTD</b>	Document Type Definition. A document type definition defines which elements are permitted within an XML document, where these elements are permitted, which attributes/attribute values are permissible, etc. Essentially, a DTD is a set of rules that an XML Parser uses to create a parse tree of a document.
<b>DTMF Tone</b>	Standard tone used to transmit numeric data via telephone. Generated when numeric keys are pressed.
<b>GPRS</b>	General Packet Radio Service. Mobile service, which gives a packet switched access over GSM to external data networks with high-peak capacity.
<b>HTML</b>	Hypertext Markup Language. The markup language used for publishing hypertext on the World Wide Web.
<b>HTTP</b>	Hypertext Transfer Protocol. An application-level protocol for transferring HTML documents on the internet.
<b>IDE</b>	Integrated Development Environment. A tool for application developers.
<b>SDK</b>	Software Development Kit. A kit for software developers that includes, for example, emulator, sample code, and reference documentation.
<b>Soft Key</b>	The phone has one or more function keys whose function depends on the state of the phone. The keys are called soft keys and have associated text on the display.
<b>URL</b>	Uniform Resource Locator. A string that identifies resources on the Web.
<b>WCSS</b>	WAP CSS. Cascading style sheet specification optimized for wireless devices.
<b>WTAI</b>	Wireless Telephony Applications Interface. Feature that provides the means to create telephony applications by using a WAE user agent with the appropriate WTAI function libraries.
<b>XHTML</b>	Extensible Hypertext Markup Language
<b>XHTML MP</b>	XHTML Mobile Profile. XML-based markup language that contains XHTML Basic and a few additional elements from full XHTML 1.1.
<b>XML</b>	eXtensible Markup Language. A World Wide Web consortium (W3C) standard for Internet markup languages. XML is a restricted subset of SGML.

---

# 1. Introduction

## 1.1 Purpose and Scope

The purpose of this document is to give an introduction to the process of creating XHTML/WML applications for the Series 60 browser. The document contains information about development tools, a simple example, and an overview of the publishing procedure.

## 1.2 Series 60 Mobile Browser

The Series 60 browser is a dual-mode browser that natively supports XHTML Mobile Profile (XHTML MP) with WAP CSS and WML 1.x. This means that existing applications written in WML 1.x will work in the browser.

The major difference between XHTML and WML is that XHTML supports the use of style sheets. XHTML MP with WAP CSS gives developers much more control over the format and presentation of WAP pages. XHTML MP and WML 1.x pages may be seamlessly linked together; history is maintained so the user can go back regardless of the content of previous pages.

## 1.3 References

Designing Mobile Services for Series 60 Browser <http://forum.nokia.com>

Nokia Mobile Internet Toolkit 3.1 <http://forum.nokia.com>

WAP Service Developer's Guide for Nokia Series 60 Phones with XHTML Browser <http://forum.nokia.com>

Series 60 Content Authoring SDK v1.0 for Symbian OS <http://forum.nokia.com>

Browsing on Mobile Devices <http://forum.nokia.com>

XHTML Mobile Profile <http://www1.wapforum.org/tech/documents/WAP-277-XHTMLMP-20011029-a.pdf>

Building effective WAP Sites and Services <http://www.imperium.co.uk>

XHTML in Mobile Application Development <http://www.soberit.hut.fi/T-121/suomi/XHTMLinMobileApplicationDevelopment.pdf>

<http://www.thewirelessfaq.com>

<http://www.mobileinfo.com>

<http://www.ericsson.com>

## **2. Creating Mobile Services for Series 60 Browser**

### **2.1 Effective XHTML/WML Applications**

While this document aims to outline the process involved in creating mobile services for the Series 60 browser, it is important to recognise that producing an effective service involves far more than writing bug-free code. The developer needs to create a service that people want to use, and which they find easy to interact with.

Because services for the mobile browser offer a very different experience to their Web-based equivalent, it is recommended that the developer spend time exploring several WAP sites to appreciate how the medium is best utilized. In particular, the developer should understand how the small screen size, limited graphical capability, and navigation techniques impact the user experience.

Once the basics have been learned, it may be tempting to leap straight into the coding and allow the application to evolve, but it is unlikely that the finished product will be satisfactory. The first step should always be to consider carefully the purpose of the application and the target audience; the project objectives should then be clearly defined.

#### **2.1.1 Content**

Content should be focused. As a guide, consider that each time the handset accesses new content it should be in response to a request for new information, rather than a continuation of the previous request. There are times when this will be inappropriate, but if the user can be given all the information he or she wants in one well-presented page, then two should not be used.

#### **2.1.2 Design**

When designing an application, it is essential to consider the user experience. Mobile devices are not commonly used for surfing and, therefore, a user's specific requirements should be anticipated and the application constructed to enable these needs to be met as quickly as possible. For example, a long list of links on a page is acceptable if this will reduce the navigation route to other pages.

It is recommended that the developer plan an application on paper before coding is started. Individual pages should be sketched out and the hierarchy built up, followed by a walk-through of all Use Cases to ensure the design is efficient.

### **2.2 Dynamic XHTML/WML Applications**

#### **2.2.1 Creating dynamic content**

A dynamic application has content that is generated at run-time. Rather than just hard-code a series of static pages for an audience to browse, server-side technologies can be used to create the pages on-the-fly. For example, a database holding information about warehouse stock could be queried by a user for specific product details and then XHTML code could be generated to display the results.

Suitable Web application servers include Microsoft's Active Server Pages (ASP), Allaire's ColdFusion and Java Server Pages (JSP), but detailed information about the use of these technologies is beyond the scope of this introductory document.

## 2.2.2 WAP Push

The power to create dynamic applications enormously increases the potential of WAP, and this was further enhanced by the introduction of WAP Push in WAP 1.2.1. This is essentially a notification process whereby content is sent out (or pushed) to a browser, rather than a browser sending a request for information. So, for example, interested parties could be informed immediately of stock deliveries or, in a turn-based game, a player could be informed of his or her opponent's move.

Push service messages indicate downloadable content. The user can view indicated content by using the Service Inbox and the WAP browser.

### 2.2.2.1 Service Indication

The Service Indication (SI) content type enables an alert message to be sent to users. A Service Indication must contain at least the text to be displayed to the user and a URL address of content that may be dynamically generated for each push message in the application server. In addition to containing text and a URL address, the Service Indication can optionally contain the following information: creation date and time, expiration date and time, ID, and priority.

### 2.2.2.2 Service Loading

The Service Loading content type enables a URL address to be sent to a device, which then "pulls" the content at that URL from the origin server without the user specifically requesting it. Because the user may be busy doing other things when service-loading content is received, the intrusiveness level can be set.

A Service Loading content type has one mandatory href attribute and one optional action attribute. The href attribute is the URL of the service application to be loaded and the action attribute specifies the action that should be taken after "pulling" the service application.



---

## 3. Development Tools

### 3.1 SDKS and IDEs

Although WAP documents can be written using a simple text editor, a number of SDKs and IDEs are available to aid the development of wireless applications. A few examples are listed below.

#### 3.1.1 Nokia Mobile Toolkit 3.1

The Nokia Mobile Internet Toolkit 3.1 (available free from [www.forum.nokia.com](http://www.forum.nokia.com)) is an application development and testing environment. It provides editors and simulators to aid the authoring, testing, and debugging of XHTML and WML mobile applications.

#### 3.1.2 Series 60 Content Authoring SDK 1.0 for Symbian OS, Nokia edition

The Series 60 Content Authoring SDK (available free from [www.forum.nokia.com](http://www.forum.nokia.com)) offers the Series 60 mobile developer an opportunity to preview how their applications will look on an actual Series 60 device. The application can be used stand-alone or can be integrated with the Nokia Mobile Internet Toolkit 3.1. However, it should be noted that the Series 60 Content Authoring SDK only offers support for WML.

#### 3.1.3 Ericsson WAP IDE 3.2.1

This offers similar functionality to the Nokia Mobile Internet Toolkit 3.1 but provides simulators for Ericsson phones and is available free from [www.ericsson.com](http://www.ericsson.com).

### 3.2 Image Tools

Utilities are available to convert images from formats such as bmp and jpeg to the wireless bitmap format (WBMP). An online conversion tool can be found at [www.teraflops.com](http://www.teraflops.com) or downloaded from [www.imagemagick.org](http://www.imagemagick.org), while other tools that offer both creation and conversion can be found at [www.imagemagick.org](http://www.imagemagick.org).

### 3.3 Other Helpful Applications

There are numerous free applications available to help the wireless developer. In addition to the SDKS and IDEs, there are many WAP emulators and validation tools for XHTML, CSS and WML that can be found on the web.

## 4. Writing A Simple XHTML MP Application

### 4.1 Application Essentials

The Series 60 browser supports XHTML MP and WML 1.x, and one of these markup languages could be used when writing a simple application for the Series 60 browser. Traditionally, WML has been the markup language used in WAP applications, but the introduction of XHTML MP, a subset of XHTML 1.1, has been widely applauded. The example below (4.2) is written in XHTML MP and a style sheet applied to control the presentation.

#### 4.1.1 XHTML MP

XHTML MP is based on XHTML and uses the same markup rules. An XHTML element consists of a start tag, element content, and end tag, written as follows:

```
<element> element content </element>
```

The following rules apply to all XHTML MP documents:

- All XHTML documents must be well formed, i.e., they conform to XML syntax rule.
- The root element must be `<html>`.
- The name of the default namespace of the root element must be the XHTML namespace name, <http://www.w3.org/1999/xhtml>.
- A single DOCTYPE declaration is required prior to the root element, with a public identifier. (This declaration gives a reference to the DTD to which the document should be compared for validation.) For example,
 

```
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN"
"http://www.wapforum.org/DTD/xhtml-mobile10.dtd"
```
- The DTD internal subset must not be used to override any parameters in the DTD.
- The elements `<head>`, `<title>`, and `<body>` are all mandatory.

#### 4.1.2 Style sheets

##### 4.1.2.1 WAP CSS

Cascading Style Sheets (CSS) were introduced to improve both the consistency and efficiency of Web-based applications; WAP CSS are the mobile equivalent. Essentially, a CSS defines how to display XHTML elements, in terms of font, alignment, spacing, etc., and should be used in partnership with XHTML documents. The XHTML should be seen as providing the structure, and the CSS the style.

A WAP CSS can be written in any text editor. An example of a very simple style sheet (styleExample.css) for XHTML MP is shown below:

```
P{font-family: Arial, sans-serif; font-size: 120%;}
```

The file should be saved with a .css extension in a directory that can be accessed by the XHTML documents. To link a XHTML page with the style sheet, a `<link>` element should be included in the `<head>` element of the document, for example,

```
<link rel="stylesheet" href="styleExample.css" type = "text/css"/>
```

#### 4.1.2.2 Internal style sheets

It is also possible to add style to documents by using the `<style>` element, positioned in the document header.

E.g.

```
<html>
<head>
<style type = "text/css">
p{ text-align: center; }
</style>
</head>
</html>
```

#### 4.1.2.3 Inline style sheets

If a style should be applied only to a particular element, an inline style can be used, as, for example:

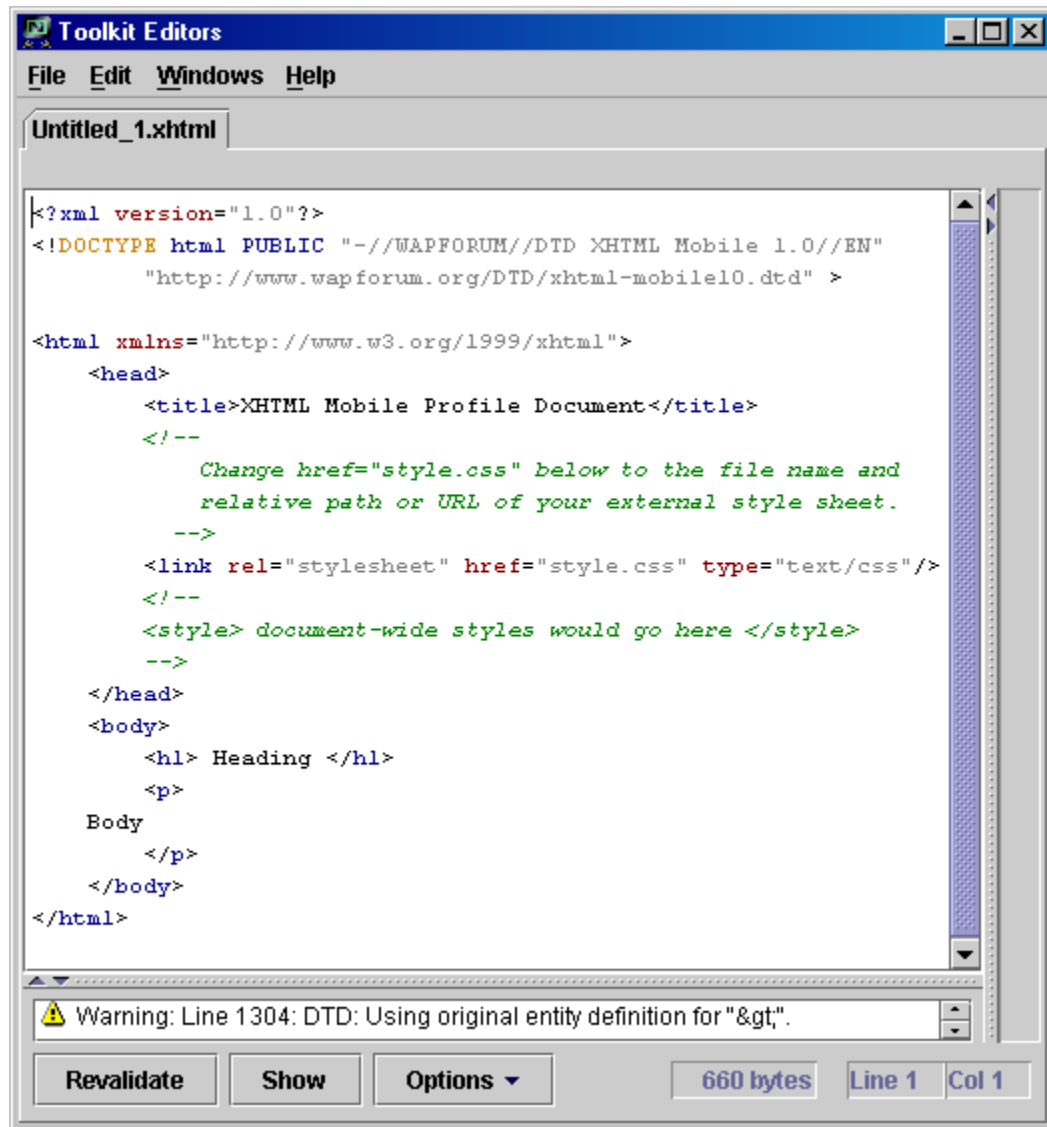
```
<p style="text-align: center">...</p>
```

## 4.2 Creating a Simple XHTML Page

Using one of the SDKs mentioned in Chapter 3 makes producing a simple XHTML MP application trivial. The Nokia Mobile Internet Toolkit 3.1 and Nokia Mobile Browser simulator are used in the example below, but the resulting XHTML document could be produced with any other of the SDKs or a simple text editor.

### 4.2.1 Using Nokia Mobile Internet Toolkit 3.1

After installing and opening Nokia Mobile Internet Toolkit 3.1, select *New* from the *File* menu and then choose the *XHTML-MP* option. A window will be displayed containing the code for a simple WAP (XHTML) page.



**Figure 4.1: Code for simple XHTML page, produced by Nokia Mobile Internet Toolkit 3.1**

The first two lines of code indicate that the document conforms to xml version 1.0 and the DTD, xhtml-mobile10. The third line sets `<html>` as the root element, and defines its namespace. These lines should be considered as the starting point for every XHTML MP document. The mandatory elements, `<head>`, `<body>`, and `<title>` should also be included and positioned as shown.

In addition to the required elements, there is a link to a style sheet (see Section 4.2.2), a heading of 'heading' and the word 'body' as paragraph (`<p>`) text.

The document can be previewed by saving the document and then pressing the *Show* button.

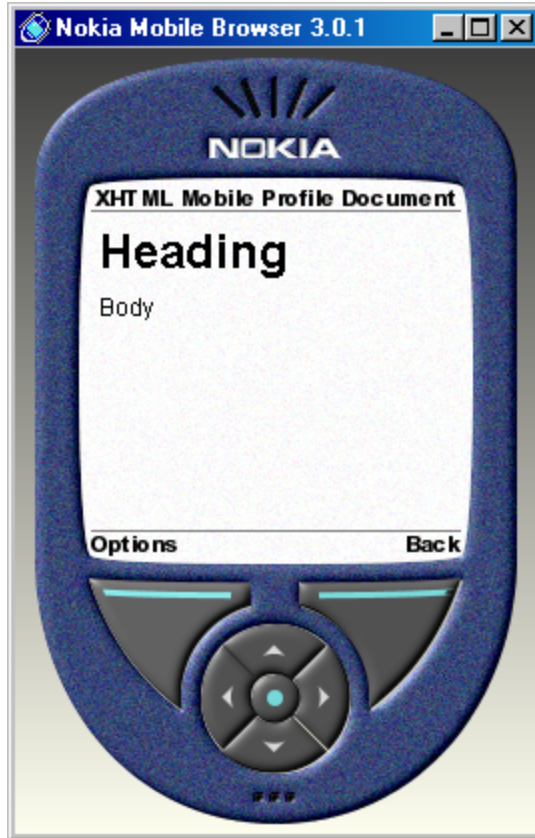
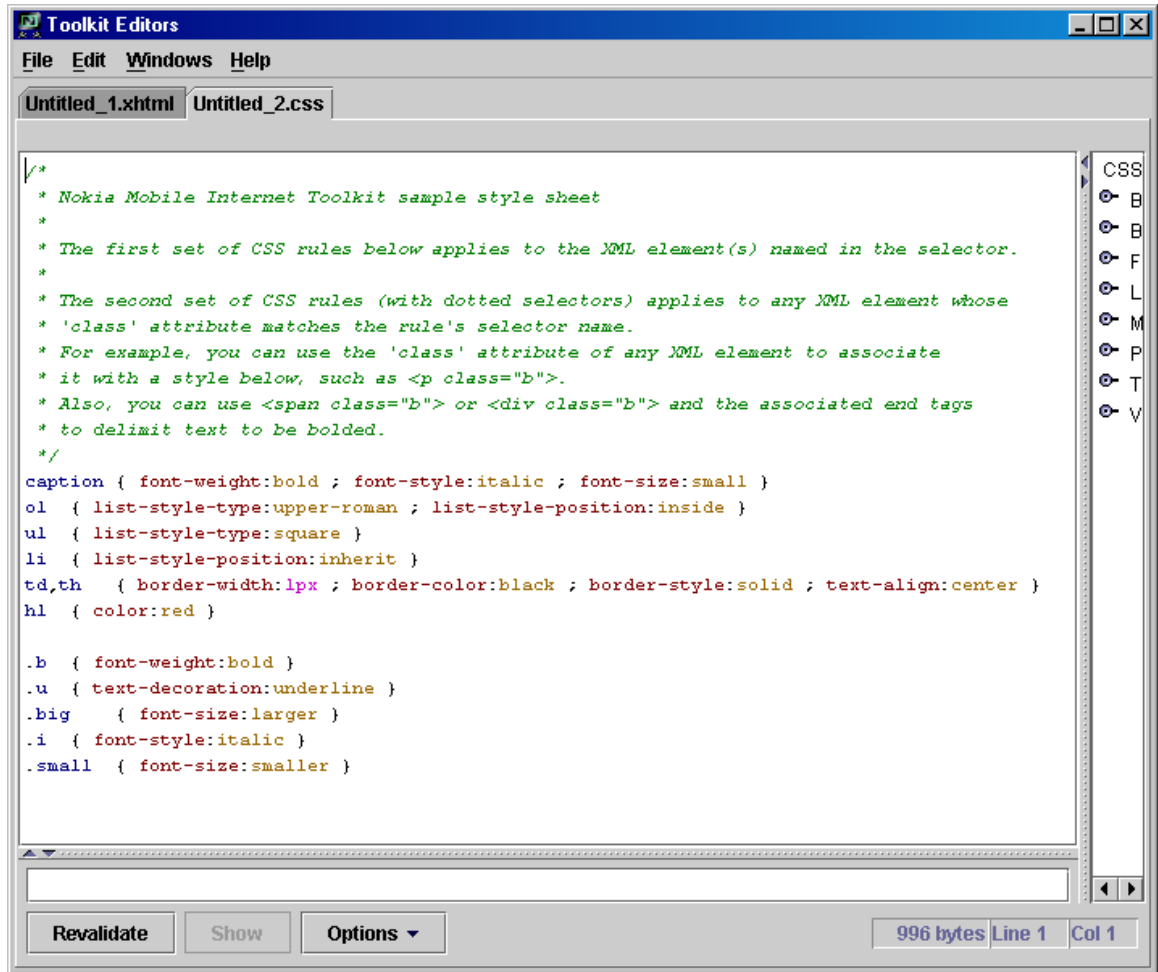


Figure 4.2: Default XHTML file, previewed in Nokia Mobile Internet Toolkit 3.1

#### 4.2.2 Adding a style sheet

When rendering the page, the line of code `<link rel="stylesheet" href="style.css" type="text/css"/>`, will cause the browser to look for the style sheet, *style.css*. If it is unable to find it, the browser will use the default styles. Because the style sheet has not yet been created, the page is displayed with the default behavior.

Creating a style sheet is also very simple. Choose *File / New* from the Toolkit menu and then select the *CSS* type. A sample style sheet will be generated, as shown in Figure 4.3. (Note that the style sheet produced has a rule to display the `h1` header in red.) Next, save the as *style.css*, ensuring that it is located in the same directory as the XHTML file.



**Figure 4.3: Default style sheet, generated by Nokia Mobile Internet Toolkit 3.1**

The effects of the style sheet can be seen by returning to the XHTML MP code and choosing to *Show* the page again.

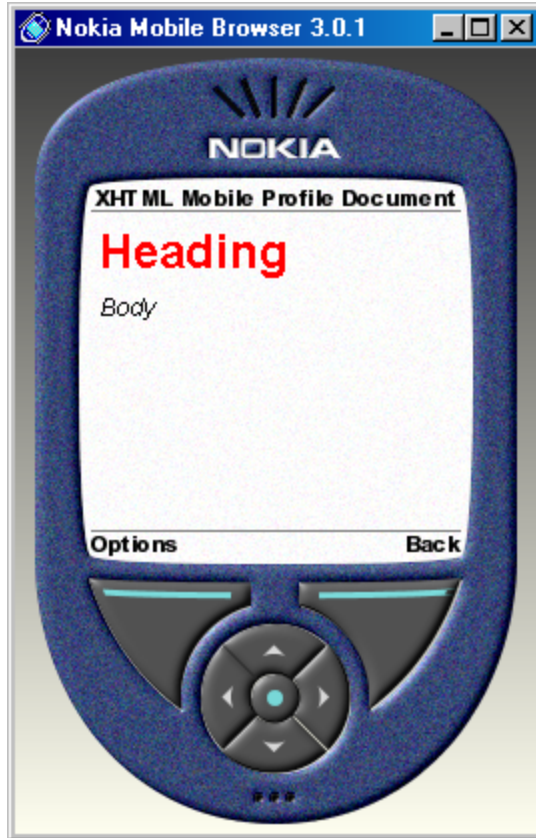


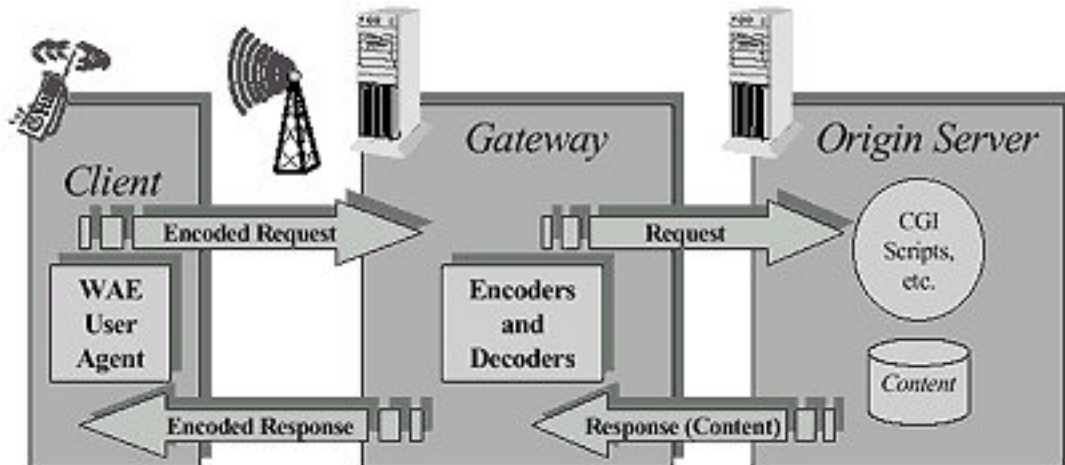
Figure 4.4: Default XHTML file with default style sheet applied, previewed in Nokia Mobile Internet Toolkit 3.1

## 5. Publishing an XHTML/WML Application

When an application has been written, it will need to be uploaded to a server so that it can be accessed globally. The mechanics for loading XHTML/WML pages by mobile browsers is described below.

### 5.1 How a Series 60 Browser Loads an Application

Before loading a URL, the Series 60 device must first establish a connection with a network provider, either via GPRS or through a dial-up server. This step provides the browser with access to the low-level protocols it needs, and the device is assigned an IP address. The request for the URL is then sent to a WAP gateway. Essentially, a WAP gateway is software that provides a link to an external network, like the Internet. It will submit a normal http request (e.g., GET <http://www.o2.ie>) to a Web server and then convert the response into a form understandable by a mobile device. When the browser receives the code from the gateway, it renders the contents on the WAP device's display.



**Figure 5.1: WAP Gateway Architecture**

An application must be resident on a server before it can be accessed by mobile devices. If a server is not available to the developer, then many ISPs offer site hosting facilities. However, with this option the developer is usually limited to static pages that will have to be updated periodically. To create dynamic applications, and to harness the full power of WAP, it is probable that the developer will need access to his or her own server, such as an Apache Server.

### 5.2 MIME Types

MIME types are required by a Web server to identify what type of file a browser is requesting, so it is necessary to furnish the server with WAP specific information.

The following list of MIME types for wireless applications is not exhaustive, but it does cover the most commonly used files.

File Type	MIME Type	Extension
<b>XHTML Mobile Profile source</b>	text/html application/xhtml+xml	.html, .htm, .xhtml



	application/vnd.wap.xhtml+xml	
<b>WML source</b>	text/vnd.wap.wml	.wml
<b>WMLScript source</b>	text/vnd.wap.wmlscript	.wmls
<b>WBMP image</b>	text/vnd.wbmp	.wbmp
<b>CSS</b>	text/css	.css
<b>Bitmap image</b>	image/bmp	.bmp
<b>JPEG image</b>	image/jpeg	.jpeg
<b>Gif image</b>	image/gif	.gif
<b>Png images</b>	image/png	.png
<b>MPEG audio</b>	audio/mpeg	.mpga, .mp2

---

## 6. Summary

Although sometimes met with skepticism, WAP technologies offer the developer a rich toolset with which to create dynamic and effective services for the mobile browser. At the same time, the simplicity of creating markup language documents and the freely available SDKs means that static applications can be created quickly and easily. Irrespective of the purpose of the service, the developer should see the design phase as critical. In summation, mobile browsing services should be well planned, with focused content and usability.

The Series 60 browser offers an excellent medium with which to exploit the latest mobile browsing technologies, as it offers support for globally accepted open standards such as XHTML Mobile Profile and WAP CSS. This offers the developer enhanced control over page design and aids the creation of consistent browsing applications. Besides this, Series 60 mobile browser also supports traditional WAP features, such as WAP Push functionality, Wireless Transport Layer Security (WTLS), and WTAI.