

WAP - ENVIRONMENT

The uppermost layer in the WAP stack, the Wireless Application Environment *WAE* provides an environment that enables a wide range of applications to be used on wireless devices. In the chapter "WAP - the wireless service enabler" the WAP WAE programming model was introduced. This chapter will focus on the various components of WAE:

- **Addressing model**

A syntax suitable for naming resources stored on servers. WAP use the same addressing model as the one used on the Internet that is Uniform Resource Locators *URL*.

- **Wireless Markup Language *WML***

A lightweight markup language designed to meet the constraints of a wireless environment with low bandwidth and small handheld devices. The Wireless Markup Language is WAP's analogy to HTML used on the WWW. WML is based on the Extensible Markup Language *XML*.

- **WMLScript**

A lightweight scripting language. WMLScript is based on ECMAScript, the same scripting language that JavaScript is based on. It can be used for enhancing services written in WML in the way that it to some extent adds intelligence to the services; for example, procedural logic, loops, conditional expressions, and computational functions.

- **Wireless Telephony Application *WTA, WTAI***

A framework and programming interface for telephony services. The Wireless Telephony Application *WTA* environment provides a means to create telephony services using WAP.

Hardware and Software Requirement:

At minimum developing WAP applications requires a web server and a WAP simulator. Using simulator software while developing a WAP application is convenient as all the required software can be installed on the development PC.

Although, software simulators are good in their own right, no WAP application should go into production without testing it with actual hardware. The following list gives a quick overview of the necessary hardware and software to test and develop WAP applications:

- A web server with connection to the Internet
- A WML to develop WAP application
- A WAP simulator to test WAP application
- A WAP gateway
- A WAP phone for final testing.

Microsoft IIS or Apache on Windows or Linux can be used as the web server and Nokia WAP Toolkit version 2.0 as the WAP simulator.

Please have look at [WAP - Useful Resources](#) to find out all the above components.

Configure Web Server for WAP:

In the WAP architecture, the web server communicates with the WAP gateway, accepting HTTP requests and returning WML code to the gateway. The HTTP protocol mandates that each reply must include something called a Multi-Purpose Internet Mail Extensions *MIME* type.

In normal web applications, this MIME type is set to text/html, designating normal HTML code. Images on the other hand could be specified as image/gif or image/jpeg for instance. With this content type specification, the web browser knows the data type that the web server returns.

In WAP applications a new set of MIME types must be used, as shown in the following table:

File type	MIME type
WML . <i>wml</i>	text/vnd.wap.wml
WMLScript . <i>wmls</i>	text/vnd.wap.wmlscript
WBMP . <i>wbmp</i>	image/vnd.wap.wbmp

In dynamic applications, the MIME type must be set on the fly, whereas in static WAP applications, the web server must be configured appropriately.

For more information about configuring MIME types for your web server, please consult your web server documentation.

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js