

SYBEX Sample Chapter

# Getting Started with WAP and WML

Huw Evans  
Paul Ashworth

## Chapter 1: Introducing WAP and WML

Copyright © 2001 SYBEX Inc., 1151 Marina Village Parkway, Alameda, CA 94501. World rights reserved. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic or other record, without the prior agreement and written permission of the publisher.

ISBN: 0-7821-2870-X

SYBEX and the SYBEX logo are either registered trademarks or trademarks of SYBEX Inc. in the USA and other countries.

TRADEMARKS: Sybex has attempted throughout this book to distinguish proprietary trademarks from descriptive terms by following the capitalization style used by the manufacturer. Copyrights and trademarks of all products and services listed or described herein are property of their respective owners and companies. All rules and laws pertaining to said copyrights and trademarks are inferred.

This document may contain images, text, trademarks, logos, and/or other material owned by third parties. All rights reserved. Such material may not be copied, distributed, transmitted, or stored without the express, prior, written consent of the owner.

The author and publisher have made their best efforts to prepare this book, and the content is based upon final release software whenever possible. Portions of the manuscript may be based upon pre-release versions supplied by software manufacturers. The author and the publisher make no representation or warranties of any kind with regard to the completeness or accuracy of the contents herein and accept no liability of any kind including but not limited to performance, merchantability, fitness for any particular purpose, or any losses or damages of any kind caused or alleged to be caused directly or indirectly from this book.

Chapter

1

# **Introducing WAP and WML**

**T**his chapter introduces you to the Wireless Application Protocol (WAP) and the key markup language you'll need to grasp to develop content for WAP-enabled wireless devices. This language, called Wireless Markup Language (WML), is supplemented by its own scripting language, called WMLScript, which will be introduced before a discussion on WAP architecture, standards, devices, and what the future holds in this fast-moving technological environment.

## What Is WAP?

*Wireless Application Protocol (WAP)* is a set of standards designed to extend Internet services to mobile phones, pagers, and personal digital assistants (PDAs). The development of WAP is coordinated by an industry-wide group of companies under the banner of the WAP Forum. The WAP Forum was created to apply the best principles of Internet application development to the wireless environment. See the sidebar later in this chapter for more on the WAP Forum.

WAP has become the de facto worldwide standard for the presentation and delivery of wireless information and telephony services on mobile phones and other wireless devices. It is an open protocol that provides the same development standards to all vendors irrespective of the underlying network system. It is designed to work under the low bandwidth constraints of wireless networks. These constraints are currently around 10Kbps compared with the standard 56Kbps on home computers using domestic telephone lines with standard dial-up modems. As most wireless computing devices have limited processing power and memory, and are designed with screen displays and small multifunction keypads, WAP was developed with these limitations in mind. WAP was created to address three main issues related to data communications across wireless networks: low bandwidth, high latency, and connection availability.

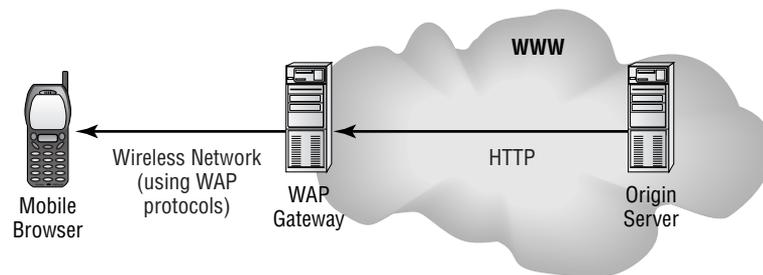
Because WAP is an open protocol, a number of manufacturers are producing a wide range of WAP-enabled devices. These manufacturers in turn are able to source from a large range of WAP-specified components because the server technology is also open. This general openness and adoption of common standards means that developers, manufacturers, and content providers are able to adopt WAP with confidence and benefit from the economies of scale.

WAP uses a client/server architecture that employs an unsophisticated wireless-based microbrowser and requires only limited resources and a WAP gateway to deliver content from the server where it is stored. It is a standard independent of the air interface, the user interface, and the underlying data bearer. Therefore, it is entirely interoperable. Because WAP is based on existing Internet technologies, it leverages the massive investment in similar conventional Web tools, applications, servers, and their developers while considering the restricted bandwidth, processing power, and memory currently available on wireless portable devices.

Web content is available over existing wireless communication networks through a WAP gateway. Figure 1.1 illustrates how an established WWW infrastructure, based on the Hypertext Transfer Protocol (HTTP), uses a WAP gateway to interface with the wireless network by translating HTTP requests into wireless device requests.

**FIGURE 1.1:**

The WAP gateway



The 1997 WAP Forum specification is unique in that it is truly global and spans the numerous airlink standards. WAP specifies guidelines for the implementation of microbrowsers and the network-resident servers that connect portable wireless devices to the network infrastructure and the Web. For the carriers, the creation of WAP means minimum risk and investment, new sources of revenue, and a more attractive, comprehensive suite of value-added services. For users, it provides easy access via WAP-enabled handheld devices to the Web and secure access to corporate intranets. Manufacturers also benefit from the provision of a global, open, de facto standard. They're provided with a complete new range of marketing opportunities and revenue streams, while the content providers and developers are confronted with the huge untapped market of wireless customers and the prospect of 100 million WAP-enabled devices by mid-2001.

## What Is WML?

*Wireless Markup Language (WML)* is to WAP and its handheld devices what Hypertext Markup Language (HTML) is to the Web and browsers such as Netscape and Internet Explorer. It describes how content is presented to the wireless device, allowing you to display information, present input options, and tell user agents (programs that interpret WML, WMLScript, and other forms of code—typically a microbrowser in a mobile phone) how to respond once an option has been selected using the keypad.

WML is a subset or an application of the eXtensible Markup Language (XML), and because WAP uses a similar model as the Internet, it allows content developers to quickly become proficient with this relatively simple tag-based language while allowing a clear development path. WML is based on the World Wide Web Consortium (W3C) guidelines for wireless access and works similarly to HTML to deliver Web text using simple markup tags.

WML's user interface is a WAP microbrowser optimized to map onto mobile wireless devices. A WML document is called a *deck*, which is comparable to an HTML page. Unlike the flat structure of HTML content, WML documents—or decks—are divided into separate units of user interaction. Each unit is called a *card*, and WAP services are created by letting the user navigate between the cards of one or more decks in much the same way hyperlinks are used within and between HTML documents.

WAP gateways provide the interface between the network and Internet or intranet services. From this gateway, WML content is accessed over the Internet using the standard HTTP mechanism.

WAP developers and content providers can get up to speed quickly with WML, as it follows the same programming model as the Web development model. It is a tag-based language specified as an XML document type, so all existing XML tools and some HTML development environments can be used to develop WAP applications. As standard HTTP is used for communication between gateway and servers, developers can use off-the-shelf Web servers to deploy their applications. Standard tools such as ColdFusion and CGI scripting languages such as Perl, PHP, and ASP generate content for dynamic WML applications.

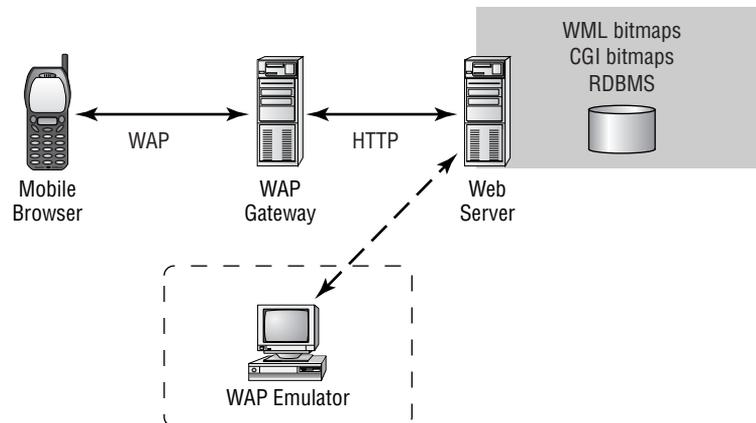
The WML specification was developed by the WAP Forum and defines the syntax, variables, and elements to be used in a valid WML file. The WML 1.1 Document Type Definition (DTD) is available from [www.wapforum.org/DTD/wml\\_1.1.xml](http://www.wapforum.org/DTD/wml_1.1.xml), and all WML applications must correspond to it. The microbrowser that all WAP-compliant wireless devices are loaded with is able to handle all entities in the WML 1.1 DTD.

The WAP gateway translates wireless device requests into HTTP requests and then redirects the Web server's HTTP responses to the device. WML files being sent to WAP-enabled handheld devices are compressed into a binary format by the WAP gateway. It is possible to translate HTML into WML using a number of available filter tools, but in practice, the differing user interfaces employed between the desktop and wireless environments make specific WML-tailored solutions the norm.

A WAP emulator is a program that implements a WAP microbrowser, but is designed to run on a non-WAP device, such as a Windows PC. Emulators are often used by developers to speed up the development process, as well as to reduce the costs of using WAP during testing each time a change is made. They are also useful to see how a real WAP device will look, because many of these emulators mimic the look and feel of real WAP devices. A list of available emulators is provided in later chapters. When a WAP emulator is used, a WAP gateway is not required because WML files are downloaded from a Web server or local file, and the emulator renders it in the Emulator window. Figure 1.2 shows this process.

**FIGURE 1.2:**

Displaying WAP content using a gateway and an emulator



Applications and services written in WML become available to all network devices that are WAP compliant, delivering a *write once, use anywhere* tool. WML developers can map soft keys (appropriated to physical keys on the WAP-enabled device by the manufacturer) for easy user input and take advantage of features designed to maximize the effect of displaying text on a limited screen. Low-resolution graphics known as WBMPs can be used, and as bandwidth increases, so too will the resolution of the graphics supported by the WAP specification. As long as a recipient device is WAP compliant, the size of the device's display is automatically accounted for by the microbrowser and the use of standard HTTP header mechanisms to learn about a device's capabilities. These header mechanisms allow the developer to customize applications to take advantage of different device characteristics. This technique is known as *device sensing* and is the same in principle as the HTML of many Web sites that detect which Web browser and version you are running to provide an optimal display and to allow support for browser-specific features.

While WML is good at things like event processing, input and output handling, and rendering, it has no practical processing abilities. It is, therefore, supplemented by the WMLScript scripting language.

## What Is WMLScript?

In addition to WML, the WAP Forum provides a scripting language called *WMLScript*. WMLScript is used alongside WML (or independently of it) to provide intelligence in the form of procedural logic to client services. WMLScript is to WML what JavaScript is to HTML. In fact, WMLScript is based on ECMAScript, a derivative of JavaScript, and has been designed to support the relatively low bandwidth restrictions applicable to handheld wireless devices.

WMLScript enhances the capabilities of WML in a number of ways. With it, you can access user agent facilities, check user input, generate local messages and dialogs, and execute user agent software.

This functionality allows you to locally program things like error messages and alerts for faster viewing, add to your wireless address book, and interrogate SIM cards.

WMLScript defines functions containing flow control logic constructs such as *while*, *if*, and *for*, which can be called from the main body of the WML program.

Even so, WMLScript still lacks basic programming features such as arithmetic functions, string handling, and WML 1.1 program interfacing capabilities. It therefore has to rely on a set of standard libraries to access functions relating to dialog presentation, floating point numbers, string conversion, and integer handling.

WMLScript is compiled by a WAP gateway into binary form to reduce the code's size and therefore minimize transfer time. It defines a byte-code representation and a byte-code interpreter for optimal utilization of current low bandwidth channels and wireless device memory restrictions. In brief, WMLScript uses the following syntax rules:

- The smallest unit of execution in WMLScript is a statement, and each statement must end with a semicolon (;).
- WMLScript is case sensitive.
- Comments can either be single-line (beginning with //) or multi-line (bracketed by /\* and \*/). This syntax is identical to that of both C++ and Java.
- A literal character string is defined as any sequence of zero or more characters enclosed within double (") or single (') quotes.
- Boolean literal values correspond to true and false.
- New variables are declared using the var keyword (i.e., var x;).
- WMLScript has no type checking done at compile-time or runtime, and no variable types are explicitly declared. Internally, the following data types are supported:
  - Boolean
  - Integer
  - Floating point
  - String
  - Invalid

You do not need to specify the type of any variable because WMLScript automatically attempts to convert between the different types as required. It is also worth noting that WMLScript is not object-oriented, so it is not possible to create your own user-defined data types programmatically.

WMLScript allows a variety of operators that support value assignment operations, arithmetic operations, logical operations, string operations, comparison operations, and array operations. WMLScript operators and expressions are nearly identical to those of the JavaScript programming language.

Although WMLScript does not support the creation of new objects via object-oriented programming, the six libraries that are provided help in the handling of many common tasks. The Lang library includes functions for data type manipulation, absolute value calculations, and random number generation. The optional Float library includes `sqrt()`, `round()`, and `pow()` functions but is supported only on those wireless devices with floating point capabilities. The String library contains a set of functions for performing string operations. Some of the functions included in this library are `length()`, `charAt()`, `find()`, `replace()`, and `trim()`. A URL library is provided for handling both absolute URLs and relative URLs. Extraction functions let you retrieve individual components from absolute and relative URLs, and include `getPath()`, `getReferer()`, and `getHost()`. A WML-Browser provides `go()`, `prev()`, `next()`, `getCurrentCard()`, and `refresh()`, where WMLScript can access the associated WML context. Finally, a Dialogs library provides a set of typical user-interface functions including `prompt()`, `confirm()`, and `alert()`.

## The WAP Forum

The Wireless Application Protocol Forum was created in 1997 to provide a worldwide open standard for the delivery of Internet-based services to wireless handheld devices. It was formed by an alliance of partners made up of telephone manufacturers, Motorola, Ericsson, Nokia, and Unwired Planet, a US software company that changed its name to Phone.com (which recently merged with Software.com to become Openwave).

Before the formation of this group, the concerned players were all working independently to address the issue of developing a schema to increase the capabilities of wireless telephony platforms. Ericsson had begun work with a protocol called ITTP (Intelligent Terminal Transfer Protocol) with the aim of making it easy to add services to wireless platforms. Nokia was working on their Smart Messaging concept, and Unwired Planet on their Handheld Device Markup Language (HDML), a language similar to HTML but focused on small-screen devices such as PDAs and mobile phones. The phone companies were also working, in conjunction with the World Wide Web Consortium (W3C), on specifications to provide wireless transmission of Internet data.

*Continued on next page*

All this proprietary-based work came to a head in 1997 when a US network operator called Omnipoint received several different responses from the different phone companies to a tender request they had issued for the supply of wireless information services. Omnipoint recommended that the various vendors get together and define a common standard. Thus the WAP Forum was born with a mission to create a global wireless specification to work across the different wireless technologies.

WML was formed by utilizing many of the concepts in Phone.com's HDML within the industry-standard framework of XML, practically guaranteeing its long-term success and ability to evolve without limitation.

The WAP Forum published the WAP 1.0 specification for product interoperability and content/application development in 1998, having based their work on Internet standards and technology. After WAP 1.0 was released, the WAP Forum was then opened up for membership by all organizations interested in the area.

As of early 2001, there were well over 600 members from the leading wireless device manufacturers, wireless operators, and software development companies. By 2003, it is predicted that over 90 percent of all mobile phones dispatched will be equipped with a WAP browser.

The goals of the WAP Forum are to bring Internet content and advanced data services to wireless portable devices; create a global open protocol that works across all wireless technologies; allow the creation of applications and content that can be scaled across a wide range of devices and wireless bearer networks; and use existing standards and technology wherever appropriate.

## The WAP Architecture

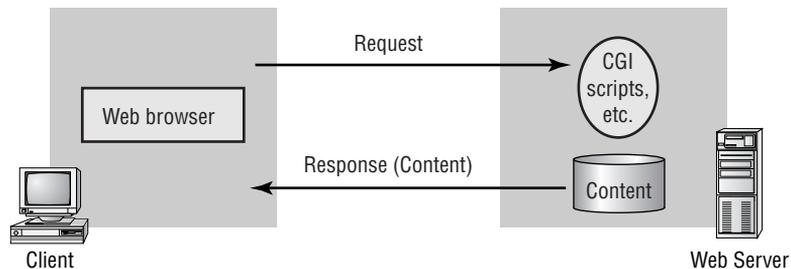
WAP's architecture applies Internet standards to microbrowser technology with the wireless device controlling how server WAP content is displayed. The protocol model is based directly on the familiar World Wide Web model, but has been optimized to provide functionality across wireless networks between wireless terminals.

The architecture is designed so that only the minimum of memory and processing resources are used by the wireless device, allowing the greatest range of low intelligence handsets to be equipped with the microbrowser. However, the architecture also allows more sophisticated devices to take advantage of higher-quality content delivery through the inclusion of animation, graphics, and scripting.

The WAP programming model is based upon the very flexible and powerful World Wide Web programming model. In this model, applications and content are presented in standard data formats and are read by Web browsers. The Web browser sends requests for data objects to a Web server, and the Web server responds with the data encoded using the standard formats. Figure 1.3 illustrates this process.

**FIGURE 1.3:**

World Wide Web programming model



The World Wide Web standards specify a standard naming model. Under this model, all servers and content are named with an Internet-standard Uniform Resource Locator (URL). It enables Web browsers to correctly process content based on a specific content type. Also defined are standard content formats such as Hypertext Markup Language (HTML) and standard protocols such as the Hypertext Transport Protocol (HTTP), which allows any Web browser to communicate with any Web server.

The World Wide Web standards also define three types of servers: an *origin server* on which given WAP content sits or is to be dynamically created; a *proxy server*, which acts both as a server and a client for the purpose of making requests on behalf of other clients; and a *gateway server*, which acts as an intermediary for some other server.

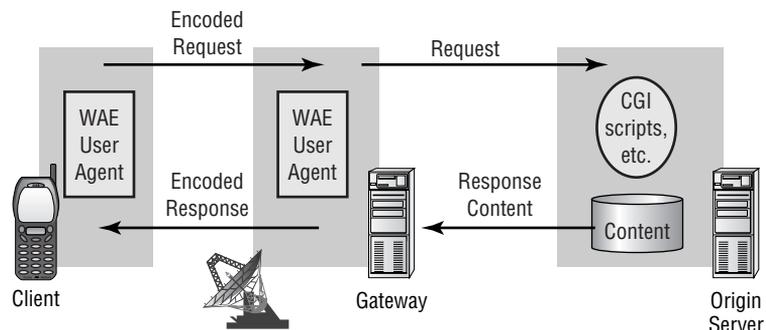
The proxy server normally resides between clients and servers that have no means of direct communication. Requests are serviced by the proxy program or passed on, with possible translation, to other servers. The proxy server implements the client and server requirements of the World Wide Web specifications. Unlike a proxy, a gateway receives requests as if it were the origin server for the requested resource. The requesting client may not be aware that it is communicating with a gateway.

The WAP programming model is simply standard Web programming with a WAP gateway in the middle of the request/response cycle. A cell phone or other wireless terminal requests, in byte code, a given URL; the WAP gateway server decodes and decompresses this, then sends it on to the appropriate origin server as an ordinary HTTP request. The process is then repeated, in reverse, on the response side of the cycle.

This programming model has ensured that application developers have a smaller learning curve with a proven architecture and the ability to leverage existing tools. The WAP programming model has had to take into account the more restricted limitations of the wireless environments, but, wherever possible, existing standards have been used or have been adopted as the starting point for the WAP technology.

Figure 1.4 illustrates this WAP programming model. A user presses a key on their WAP phone. The key has a URL request assigned to it, which the client (user agent) passes to the WAP gateway using the WAP protocol. The WAP gateway then creates a normal HTTP request for the specified URL and passes it on to the origin server for processing. The URL may refer to a static file or some form of script application. If a static file has been referenced, the origin server adds an HTTP header to the file. If a script application has been specified, then the origin server runs the application. The origin server then returns the WML document (deck) incorporating the HTTP header or any output resulting from a script application. The WAP gateway then verifies the HTTP header and the WML content, encodes them into binary format, and creates a WAP response containing the WML that it sends to the client. Upon receipt of this response, the client processes it and displays the first card of the WML deck to the user.

**FIGURE 1.4:**  
The WAP programming model



So, WAP applications and content use formats based upon World Wide Web formats, and a set of standard communication protocols is used to transport content. The WAP microbrowser acts in the same way as a Web browser, coordinating the user interface.

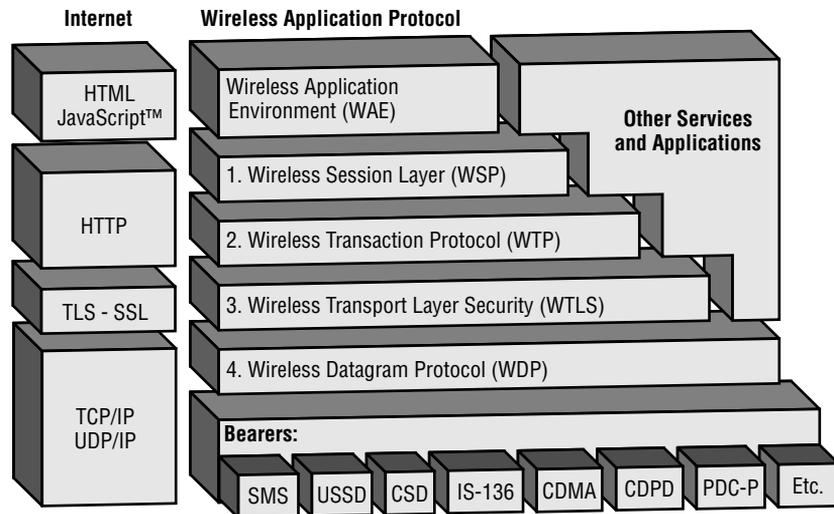
WAP uses standard World Wide Web URLs to identify WAP content on origin servers and identify local resources in a device. It uses content typing consistent with World Wide Web typing, allowing user agents to process the content based on its type, and standard World Wide Web-based content formats for display, calendar, electronic business-card objects, images, and a scripting language. It also uses standard communication protocols, allowing the communication of browser requests from the wireless terminal to the network Web server.

## The WAP Protocol

WAP is a layered communications protocol, an implementation of which is embedded in all WAP-enabled user agents. Its structure is very similar to the well-established International Standards Organization (ISO) network model with a transport protocol similar to the generally fixed-line HTTP. However, in this case, it is focused on broadcast requirements, which use less bandwidth. The WAP protocol architecture is shown in Figure 1.5 alongside a typical Internet Protocol stack. It consists of layers, which describe and specify the application/browser (WAE), sessions (WSP), transactions (WTP), security (WTLS), transports (WDP), and bearers (SMS, USSD, CSD, IS-136, CDMA, CDPD, PDC-P, etc.).

**FIGURE 1.5:**

WAP and Internet Protocol stacks



Following are brief descriptions of the main features of the Wireless Application Protocol layers.

## **Wireless Application Environment (WAE)**

The *WAE* is the top layer of the WAP stack and is of most interest to content developers because it contains, among other things, device specifications and the content development programming languages, WML and WMLScript. It is an application environment that is based on a combination of mobile telephony technologies and the World Wide Web. The purpose of the WAE is to establish an environment to build applications and services. The WAE includes a micro-browser environment that defines how the wireless device interprets and presents WML and WMLScript. It also contains components that specify the following:

- WML for creating WAP applications
- WMLScript to enhance the logic capabilities of WML
- Wireless Telephony Application Interface (WTAI), which provides telephony services for WML decks running on phone-based devices
- Content formats that define a set of data formats, including images, phone book records, and calendar information

WAE depends upon a WAP-compatible proxy server to translate between WAP and HTTP transactions and WAP and Internet Protocols.

## **Wireless Session Protocol (WSP)**

The *WSP* is a sandwich layer that links the WAE to two session services: one connection-oriented service that operates above the Wireless Transaction Protocol and one connectionless service operating above the Wireless Datagram Protocol. It is basically a binary-formatted tokenized version of HTTP, designed to provide low bandwidth browser handling on long latency networks. Unlike HTTP, WSP has been designed by the WAP Forum to provide fast connection suspension and reconnection. It has also been designed to provide content push capabilities that allow unsolicited transmission of data to user agents, which in turn allows WAP device users to be alerted, for example, to incoming e-mails, telephone calls, and faxes.

## Wireless Transaction Protocol (WTP)

The *WTP* runs on top of a datagram service such as User Datagram Protocol (UDP) and is part of the standard suite of TCP/IP protocols used to provide a simplified protocol suitable for low bandwidth wireless stations. It offers three classes of transaction service: unreliable one-way request, reliable one-way request, and reliable two-way request response. WTP supports protocol data unit concatenation and delayed acknowledgment to help reduce the number of messages sent, and to attempt to optimize the user experience by providing the information that is needed when it is needed.

## Wireless Transport Layer Security (WTLS)

*WTLS* incorporates security features that are based upon the established Transport Layer Security (TLS) protocol standard. It includes data integrity checks, privacy, service denial, and authentication services. Developers can access WTLS by using HTTPS instead of HTTP in the URL.

## Wireless Datagram Protocol (WDP)

The *WDP* allows WAP to be bearer-independent by adapting the transport layer of the underlying bearer. The *WDP* presents a consistent data format to the higher layers of the WAP protocol stack, thereby offering the advantage of bearer independence to application developers.

## Bearers

Below the *WDP* sit all of the *bearer* networks. These include Short Message Service (SMS), a facility for sending short messages; Unstructured Supplementary Service Data (USSD); Code Division Multiple Access (CDMA), for the reuse of scarce radio resources in adjacent areas; and Cellular Digital Packet Data (CDPD).

Through the Internet Protocol stack, the WAP client communicates with the WAP gateway, which sits between the wireless carrier's network on one side and the public Internet or corporate intranet on the other. Gateways can be located within carrier or corporate firewalls or both. In addition to taking care of various housekeeping tasks such as keeping track of the WAP client's bookmarks and managing its cache, the WAP server handles the interface between the two sets of network protocols, wireless (WAP) and wired (TCP/IP).

## WAP Standards

As much as possible, WAP uses existing Internet standards for the basis of its own architecture and is designed to allow standard Internet servers to provide services to wireless devices. However, Internet standards such as HTML, HTTP, TLS, and TCP, which require large amounts of mainly text-based data to be sent, are inefficient over wireless networks. Traditional HTML content cannot be displayed well on the small-sized displays of mobile phones and pagers, and navigation around and between screens is not easy in one-handed mode. So Internet standards such as HTML, HTTP, TCP, and TLS are not appropriate for the restrictions associated with wireless networks.

WAP does, however, use many other Internet standards such as eXtensible Markup Language (XML), User Datagram Protocol (UDP), and Internet Protocol (IP) to communicate with wireless devices. So WAP is based on familiar standards such as HTTP and TLS, but has been optimized for the constraints of the wireless environment. For example, a WAP gateway is required to communicate with other Internet nodes using HTTP, and the WAP specification requires devices to use standard URL addressing to request services.

So WAP has been optimized with the restrictions of the wireless environment in mind. It is designed for low bandwidth and long latency, and uses binary transmission for greater compression of data. WAP sessions deal with intermittent coverage and operate using IP over a large variety of wireless transports whenever possible.

It is important that WAP standards complement existing standards. For example, instead of the WAP specification designating how data is transmitted over the air interface, it is designed to sit on top of existing standards so that the bearer's standard is used with the WAP protocol.

The WAP Forum works closely with the World Wide Web Consortium and other bodies such as the European Telecommunications Standards Institute (ETSI), the Cellular Telecommunications Industry Association (CTIA), and the Internet Engineering Task Force (IETF) to ensure that the future versions of HTML, HTTP, and TCP take the special needs of wireless devices into account and can be supported in the WAP framework. The WAP Forum also works closely with these bodies and others as they become members of the Forum to address the enhanced capabilities of third generation (3G) wireless networks expected to emerge around 2003 and 2004. Some countries, such as Japan and Finland, plan to have 3G services in use as early as 2001.

When the WAP Forum identifies a new area of technology with no existing standards specification, it works to submit its specification to related standards groups, believing that active participation leads to the best standards. With this approach, the WAP Forum hopes to produce open, not proprietary, standards through industry consensus and with no one vendor receiving favorable treatment. To date, the WAP Forum's membership role of contributing companies stands at over 100 who subscribe to the notion of the development of open standards as the best way to produce solutions for wireless Internet access.

## WAP Devices

As well as being air-interface independent, the WAP specification is also device independent, specifying the minimum functionality a device must have, but designed to accommodate any functionality above that minimum.

A *WAP device* is a combination of hardware and software capable of running a WAP-compliant microbrowser such as a WAP-enabled mobile phone or a PDA. The use of proxy technology and compression in the network interface reduces the processing load so that inexpensive CPUs can be used in the handset, further reducing power consumption and extending battery life.

A PC can also be used as a WAP device if you download a WAP phone emulator from one of the developer sites. The emulator allows you to use a *virtual phone* on your desktop. Some major suppliers, such as Ericsson, Nokia, and Openwave, have developer sites where you can download software development kits (SDKs) containing WAP emulators.

A WAP phone can run any WAP application in the same way that a Web browser can run any HTML application. Once you have a WAP phone, you can access the Internet simply by entering URLs (or clicking bookmarked ones) and following the links that appear.

Using these devices, easy and secure access to Internet content and services such as banking, leisure, and unified messaging is made available. Furthermore, access is not restricted to the Internet; WAP devices will be able to deal with intranet information in the same way as Internet content because both are based upon HTML. This gives corporations the means of offering their employees information related to their work at any time, while on the move. This may also be

extended to the corporation's customers and suppliers by making restricted access to company extranets available, thereby opening up the supply chain to interrogation and analysis. Of course, such corporate information will need to be reconfigured into a format suitable for small-screen viewing by reprogramming in WML or passing the pages through special HTML-to-WML filters.

With the wide range of WAP-enabled wireless devices now hitting the market, users will have significant options available to them when purchasing terminals and the applications they support. Following is a selection of WAP phones that have been announced recently:

**Nokia 7110 and Nokia 7610** These two mobile phones are physically identical. The Nokia 7110 is targeted at the 900/1800MHz GSM market (Europe, Africa, Asia Pacific), whereas the Nokia 7610 is targeted at the 900/1800MHz TDMA market (USA). The user agent in both models is Nokia's WAP1.1 microbrowser. Both phones support WTLS.

**Motorola Timeport P7389** The Timeport is a tri-band 900/1800/1900MHz GSM mobile phone that can be used in much of the world. It uses the UP.Browser 4.x, supporting not only WML and WMLScript 1.1, but also HDML 3. The Timeport supports WTLS.

**Mitsubishi Trium Geo** The Trium Geo was the first pre-pay WAP phone available. It's marketed by the UK carrier BT Cellnet and is notable for allowing the user to optionally disable WMLScript processing. Like most of the initial WAP phones, it does not support WTLS.

**Sony CMD-Z5** Sony's Z5 is a dual-band 900/1800MHz GSM mobile phone containing Microsoft's Mobile Explorer, which is both a WML microbrowser and an HTML browser. It, too, does not support WTLS.

Device manufacturers can guarantee that their applications were developed to run on equipment that has the minimum functionality specified by the WAP specification while adding features that go beyond this minimum.

Because the WAP specification is so open, the services available to these devices will be wide-ranging in their nature. Information will be available via both push and pull technologies, and users will be able to interact with services by both voice and data. However, as the screen sizes and current speed and processing limitations make the types of services offered restrictive, it's unlikely that we'll see current Web-browsing patterns replicated via WAP-enabled devices until

bandwidth capabilities significantly increase. Short of an increase in bandwidth, the user-interface consideration is unlikely to change in the short term as manufacturers strive to design devices that are light and fit comfortably in the palm of the hand. Therefore, the type of information service that content providers will offer in the short term will be real-time, specific services such as news, weather, stock prices, and ticketing.

## The Future

In the next few years, mobile phones will start to benefit from very high bandwidth capabilities. Two-and-a-half and third generation communications systems (2.5G/3G) will allow enhanced services such as full-motion video images, multimedia, high fidelity sound, and fast access to the Internet. The 2.5G/3G systems will allow much higher capacity and data rates than can be offered by the restricted bandwidth currently available.

These wireless devices will be supported by a number of emerging technologies, including the following:

**GPRS (General Packet Radio System)** A packet-switched wireless protocol with transmission rates from 115Kbps to 171Kbps. It will be the first service available to offer full instant wireless access to the Web. It will require new handsets to support the higher data rates. A main benefit is that users are always connected and online, and will be charged only for the amount of data that is transported. A user can make and receive voice calls while at the same time downloading data. For GSM providers, this new technology will increase data rates of both circuit switching (High Speed Circuit Switched Data [HSCSD]) and packet switching (GPRS) by a factor of 10 to 15 times.

**EDGE (Enhanced Data Rate for GSM Evolution)** A higher bandwidth version of GPRS with speeds of up to 384Kbps, or twice that available from GPRS alone. It evolved from GSM, which is the prevailing standard throughout Europe and the Asia Pacific region—some 50 percent of the world's population. For GSM providers, this new technology will increase data rates of both circuit switching (HSCSD) and packet switching (GPRS) by a factor of 20 to 30 times.

**HSCSD (High Speed Circuit Switched Data)** A new high-speed implementation of GSM data techniques. It uses four radio channels simultaneously and will enable users to access the Internet via the GSM network at very much higher data rates than at present. Data rates can be transmitted at 38.4Kbps or even faster over GSM networks.

**UMTS (Universal Mobile Telecommunications System)** UMTS will allow a future mass market for high-quality wireless multimedia communications that will approach two billion users worldwide by the year 2010. Scheduled to launch commercially in 2002, it is set to be the technology that will allow the emergence of a new wireless Information Society. It will deliver low-cost, high-capacity wireless communications, offering data rates of 1Mbps to 2Mbps with global roaming and other advanced UMTS services.

**Bluetooth** A specification for short-range radio links between wireless devices (a low-power radio technology being developed to replace cables and infrared links with wireless transceivers fitting onto a single chip). Devices such as printers, desktops, mobile phones, and PDAs will use the technology, and it has the potential to be used for wireless LANs. It is a de facto standard with a throughput of around 1Mbps, and delivers opportunities for rapid ad hoc connections and the possibility of automatic connections between devices. For example, a user could walk into their office and have their mobile phone's address book automatically synchronize with their PC's address book. Bluetooth is considered complementary to WAP technology, and when used together, they open up an unlimited variety of applications, such as long-distance (WAP) remote control of home and office devices.

Note that all of these emerging technologies center around significant increases in the bandwidth available to wireless devices.

So what is the future for WAP? It has been designed to be independent of the underlying network technology. The original constraints WAP was designed for—intermittent coverage, small screens, low power consumption, wide scalability over bearers and devices, and one-handed operation—are still valid in 2.5G and 3G networks.

The rapid growth of the use of the Internet and the huge growth of the use of wireless technologies are creating a demand for wireless access to the Internet, intranets, and other data networks. People want at least the same speed and

power as they can currently achieve when using the Web via a desktop (and more) as they are offered connection technologies such as ISDN, with bandwidths of around 64/128Kbps, and the potential high-speed connections that will be available with the introduction of much faster services such as GPRS, EDGE, and eventually 3G systems.

What attracts the network operators is that WAP is a low-cost network upgrade that can be integrated seamlessly with no downtime on the network infrastructure. And *bridging* 2.5G technologies such as GPRS can be upgraded to 3G standards, ensuring that an operator's investment is protected. In the meantime, we must live with the restrictions, and it is reasonable to expect WAP to develop into optimized support for multimedia applications and continue to be relevant over the short and medium term.

In this chapter, you've been introduced to some key features such as WAP, WML, WMLScript, and the WAP architecture. WAP standards and devices have also been discussed together with a look at what the future holds in this quickly moving market. The next chapter, "Getting Started," will focus on WML and get you going with your first WML applications, showing you the tools you'll need to develop, implement, view, and test WML applications.