

WAP - Introduction

[WAP is] the de facto worldwide standard for providing Internet communications and advanced telephony services on digital mobile phones, pagers, personal digital assistants, and other wireless terminals - *WAP Forum*.

WAP stands for **Wireless Application Protocol**. Per the dictionary definition for each of these words we have:

- **Wireless:** Lacking or not requiring a wire or wires pertaining to radio transmission.
- **Application:** A computer program or piece of computer software that is designed to do a specific task.
- **Protocol:** A set of technical rules about how information should be transmitted and received using computers.

WAP is the set of rules governing the transmission and reception of data by computer applications on or via wireless devices like mobile phones. WAP allows wireless devices to view specifically designed pages from the Internet using only plain text and very simple black-and-white pictures.

WAP is a standardized technology for cross-platform, distributed computing very similar to the Internet's combination of Hypertext Markup Language (HTML) and Hypertext Transfer Protocol (HTTP), except that it is optimized for:

- low-display capability
- low-memory
- low-bandwidth devices, such as personal digital assistants (PDAs), wireless phones, and pagers.

WAP is designed to scale across a broad range of wireless networks like GSM, IS-95, IS-136, and PDC.

Who is behind WAP?

The Wireless Application Protocol (WAP) is a result of joint efforts taken by companies teaming up in an industry group called WAP Forum (www.wapforum.org).

On June 26, 1997, Ericsson, Motorola, Nokia, and Unwired Planet took the initiative to start a rapid creation of a standard for making advanced services within the wireless domain a reality. In December 1997, WAP Forum was formally created

and after the release of the WAP 1.0 specifications in April 1998, WAP Forum membership was opened to all.

The WAP Forum now has over 500 members and represents over 95 percent of the global handset market. Companies such as Nokia, Motorola and Ericsson are all members of the forum.

The objective of the forum is to create a license-free standard that brings information and telephony services to wireless devices.

Why is WAP Important?

Until the first WAP devices emerged, the Internet was a Internet and a mobile phone was a mobile phone. You could surf the Net, do serious research, or be entertained on the Internet using your computer, but this was limited to your computer.

Now with the appearance of WAP, the scene is that we have the massive information, communication, and data resources of the Internet becoming more easily available to anyone with a mobile phone or communications device.

WAP being open and secure, is well suited for many different applications including, but not limited to stock market information, weather forecasts, enterprise data, and games.

Despite the common misconception, developing WAP applications requires only a few modifications to existing web applications. The current set of web application development tools will easily support WAP development, and in the future more development tools will be announced.

WAP Microbrowser:

To browse a standard internet site you need a web browser. Similar way to browse a WAP enabled website, you would need a micro browser. A Micro Browser is a small piece of software that makes minimal demands on hardware, memory and CPU. It can display information written in a restricted mark-up language called WML. Although, tiny in memory footprint it supports many features and is even scriptable.

Today, all the WAP enabled mobile phones or PDAs are equipped with these micro browsers so that you can take full advantage of WAP technology.

WAP - Key Features

There are listed some of the key features offered by WAP:

A programming model similar to the Internet's:

Though WAP is a new technology, but it reuse the concepts found on the Internet. This reuse enables a quick introduction of WAP-based services, since both service developers and manufacturers are familiar with these concepts today.

Wireless Markup Language (WML):

You must be using HTML language to develop your web-based application. Same way, WML is a markup language used for authoring WAP services, fulfilling the same purpose as HTML does on the Web. In contrast to HTML, WML is designed to fit small handheld devices.

WMLScript:

Once again, you must be using Java Script or VB script to enhance the functionality of your web applications. Same way, WMLScript can be used to enhance the functionality of a service, just as Java script can be utilized in HTML. It makes it possible to add procedural logic and computational functions to WAPbased services.

Wireless Telephony Application Interface (WTAI):

The WTAI is an application framework for telephony services. WTAI user agents are able to make calls and edit the phone book by calling special WMLScript functions or by accessing special URLs. If one writes WML decks containing names of people and their phone numbers, you may add them to your phone book or call them right away just by clicking the appropriate hyperlink on the screen.

Optimized protocol stack:

The protocols used in WAP are based on well-known Internet protocols, such as HTTP and Transmission Control Protocol (TCP), but they have been optimized to address the constraints of a wireless environment, such as low bandwidth and high latency.

WAP - The Model

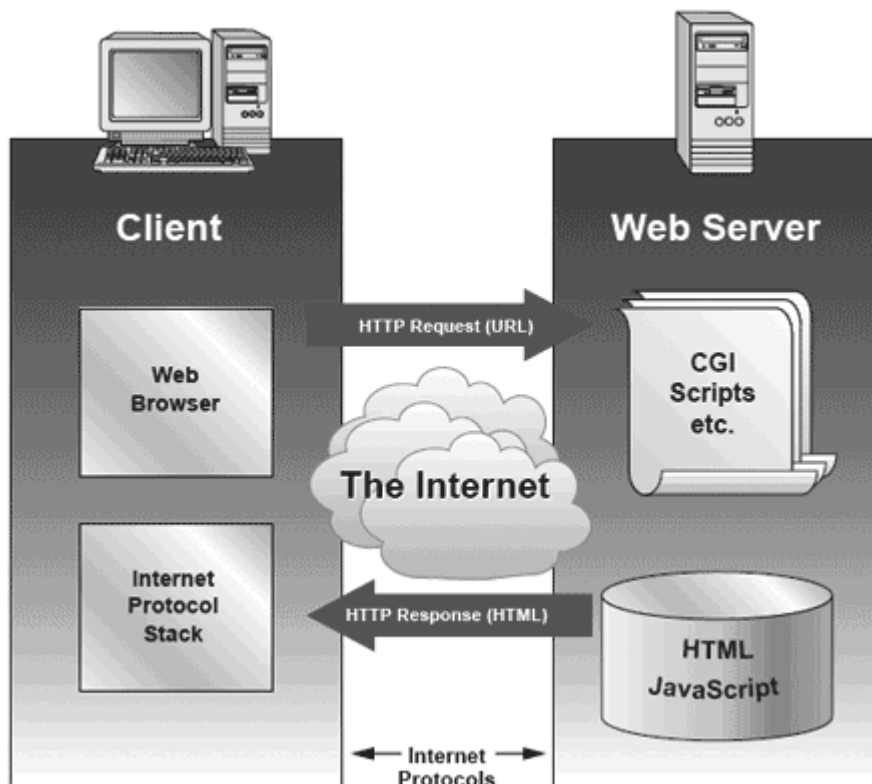
Before we describe WAP model, first we would like you to understand how Standard Internet works.

The Internet Model:

The Internet model makes it possible for a client to reach services on a large number of origin servers, each addressed by a unique Uniform Resource Locator (URL).

The content stored on the servers is of various formats, but HTML is the predominant. HTML provides the content developer with a means to describe the appearance of a service in a flat document structure. If more advanced features like procedural logic are needed, then scripting languages such as JavaScript or VB Script may be utilised.

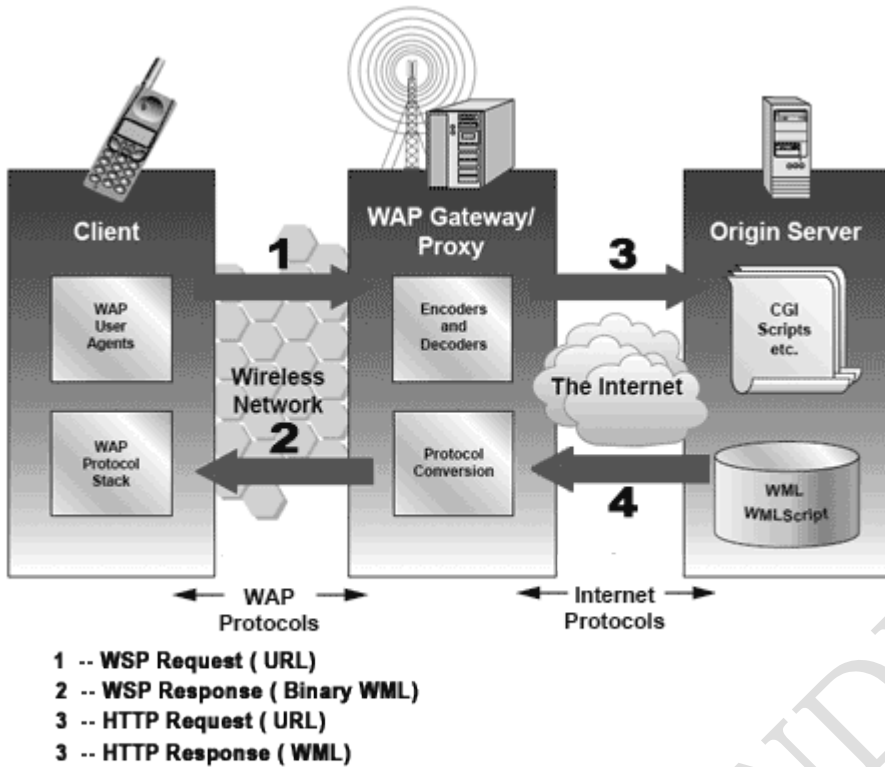
The figure below shows how a WWW client request a resource stored on a web server. On the Internet standard communication protocols, like HTTP and Transmission Control Protocol/Internet Protocol (TCP/IP) are used.



The content available at the web server may be static or dynamic. Static content is produced once and not changed or updated very often; for example, a company presentation. Dynamic content is needed when the information provided by the service changes more often; for example, timetables, news, stock quotes, and account information. Technologies such as Active Server Pages (ASP), Common Gateway Interface (CGI), and Servlets allow content to be generated dynamically.

The WAP Model:

The figure below shows the WAP programming model. Note, the similarities with the Internet model. Without the WAP Gateway/Proxy, the two models would have been practically identical.



WAP Gateway/Proxy is the entity that connects the wireless domain with the Internet. You should make a note that the request that is sent from the wireless client to the WAP Gateway/Proxy uses the Wireless Session Protocol (WSP). In its essence, WSP is a binary version of HTTP.

A markup language - the Wireless Markup Language (WML) has been adapted to develop optimized WAP applications. In order to save valuable bandwidth in the wireless network, WML can be encoded into a compact binary format. Encoding WML is one of the tasks performed by the WAP Gateway/Proxy.

How WAP Model Works?

When it comes to actual use, WAP works like this:

- The user selects an option on their mobile device that has a URL with Wireless Markup language (WML) content assigned to it.
- The phone sends the URL request via the phone network to a WAP gateway using the binary encoded WAP protocol.
- The gateway translates this WAP request into a conventional HTTP request for the specified URL and sends it on to the Internet.
- The appropriate Web server picks up the HTTP request.

- The server processes the request just as it would any other request. If the URL refers to a static WML file, the server delivers it. If a CGI script is requested, it is processed and the content returned as usual.
- The Web server adds the HTTP header to the WML content and returns it to the gateway.
- The WAP gateway compiles the WML into binary form.
- The gateway then sends the WML response back to the phone.
- The phone receives the WML via the WAP protocol.
- The micro-browser processes the WML and displays the content on the screen.

WAP - Architecture

WAP is designed in a layered fashion, so that it can be extensible, flexible, and scalable. As a result, the WAP protocol stack is divided into five layers:

- **Application Layer**

Wireless Application Environment (WAE). This layer is of most interest to content developers because it contains among other things, device specifications, and the content development programming languages, WML, and WMLScript.

- **Session Layer**

Wireless Session Protocol (WSP). Unlike HTTP, WSP has been designed by the WAP Forum to provide fast connection suspension and reconnection.

- **Transaction Layer**

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.

- **Security Layer**

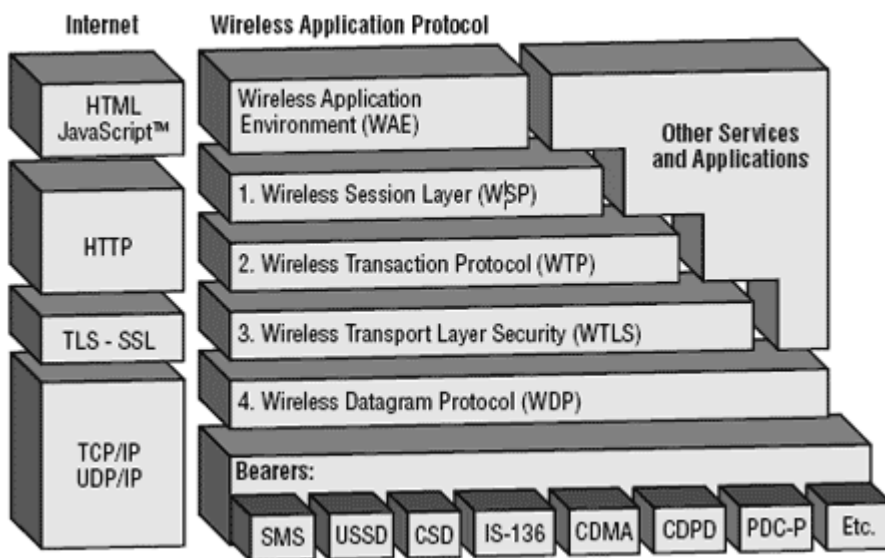
Wireless Transport Layer Security (WTLS). WTLS incorporates security features that are based upon the established Transport Layer Security (TLS) protocol standard. It includes data integrity checks, privacy, service denial, and authentication services.

- **Transport Layer**

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.

Each of these layers provides a well-defined interface to the layer above it. This means that the internal workings of any layer are transparent or invisible to the layers above it. The layered architecture allows other applications and services to utilise the features provided by the WAP-stack as well. This makes it possible to use the WAP-stack for services and applications that currently are not specified by WAP.

The WAP protocol architecture is shown below alongside a typical Internet Protocol stack.



Note that the mobile network bearers in the lower part of the figure above are not part of the WAP protocol stack.

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 WinWAP 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 (.wml)	text/vnd.wap.wml
WMLScript (.wmls)	text/vnd.wap.wmlscript
WBMP (.wbmp)	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.

WAP - WML Syntax

The topmost layer in the WAP architecture is made up of WAE (Wireless Application Environment), which consists of WML and WML scripting language.

WML scripting language is used to design applications that are sent over wireless devices such as mobile phones. This language takes care of the small screen and the low bandwidth of transmission. WML is an application of XML, which is defined in a document-type definition.

WML pages are called decks. They are constructed as a set of cards, related to each other with links. When a WML page is accessed from a mobile phone, all the cards in the page are downloaded from the WAP server to mobile phone showing the content.

WML commands and syntaxes are used to show content and to navigate between the cards. Developers can use these commands to declare variables, format text, and show images on the mobile phone.

WAP Program Structure:

A WML program is typically divided into two parts: the document prolog and the body. Consider the following code:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
<card>

...

</card>
...more cards...
</wml>
```

The first line of this text says that this is an XML document and the version is 1.0. The second line selects the document type and gives the URL of the document type definition (DTD). This DTD gives the full XML definition of WML. The DTD referenced is defined in WAP 1.1, but this header changes with the versions of the WML. The header must be copied exactly so that the tool kits automatically generate this prolog.

The body is enclosed within a `<wml>...</wml>` tag pair as shown above. The body of a WML document can consist of one or more of the following:

- Deck
- Card
- Content to be shown
- Navigation instructions

WML Commands:

The commands used in WML are summarized as follows:

Formatting:

Command	Description
---------	-------------

<p>	Paragraph
	Bold
<big>	Large
	Emphasized
<I>	Italicized
<small>	Small
	Strongly Emphasized
<u>	Underlined
 	Line Break

Inserting images:

```

```

Using Tables:

Command	Description
<table>	Definition of a table
<tr>	Defining a row
<td>	Defining a column
<thead>	Table header

Variables:

Declared as:

```
<setvar name="x" value="xyz" />
```

Used as:

\$ identifier or

\$ (identifier) or

\$ (Identifier; conversion)

Forms:

Command	Description
<select>	Define single or multiple list
<input>	Input from user
<option>	Defines an option in a selectable list
<fieldset>	Defines a set of input fields
<optgroup>	Defines an option group in a selectable list

Task Elements

Command	Description
<go>	Represents the action of switching to a new card
<noop>	Says that nothing should be done
<prev>	Represents the action of going back to the previous card

<refresh>	Refreshes some specified card variables.
-----------	--

Events:

The various events are as follows:

Command	Description
<do>	Defines a do event handler
<onevent>	Defines an onevent event handler
<postfield>	Defines a postfield event handler
<ontimer>	Defines an ontimer event handler
<onenterforward>	Defines an onenterforward handler
<onenterbackward>	Defines an onenterbackward handler
<onpick>	Defines an onpick event handler

Sample WML Program:

Keep the following WML code into info.wml on your server. If your server is WAP enabled then you can access this page using any WAP device.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<!-- WML prolog.declaration of file type and version>

<wml>
<!-- Declaration of the WML deck>
<card id="info" newcontext="true">
```

```
<!-- declaration of a card in deck>
<p align="center"><b>Information Center</b></p>
<!--paragraph declaration to display heading>
<p>
<!--paragraph declaration to display links>
<a href="Movie.wml">1. Movies info.</a>
<a href="Weather.wml">2. Weather Info.</a>
<!--declaration of links for weather and movies>
</p>
</card>
<!-- card end>
</wml>
<!-- program end>
```

WAP - WML Script

WMLScript (Wireless Markup Language Script) is the client-side scripting language of WML (Wireless Markup Language). A scripting language is similar to a programming language, but is of lighter weight. With WMLScript, the wireless device can do some of the processing and computation. This reduces the number of requests and responses to/from the server.

This chapter will give brief description of all the important WML Script components.

WML Script Components:

WML Script is very similar to Java Script. Almost WML Script components have similar meaning as they have in Java Script. A WML Script program components are summarized as follows:

WML Script Operators:

WML Script supports following type of operators.

- Arithmetic Operators
- Comparison Operators
- Logical (or Relational) Operators
- Assignment Operators

- Conditional (or ternary) Operators

Check for complete detail of [The WML Operators](#).

WML Script Control Statements:

Control statements are used for controlling the sequence and iterations in a program.

Statement	Description
if-else	Conditional branching
for	Making self-incremented fixed iteration loop
while	Making variable iteration loop
break	Terminates a loop
continue	Quit the current iteration of a loop

Check for complete detail of [WML Script Control Statements](#).

WML Script Functions:

The user-defined functions are declared in a separate file having the extension .wmls. Functions are declared as follows:

```
function name (parameters)
{
    control statements;
    return var;
}
```

The functions used are stored in a separate file with the extension .wmls. The functions are called as the filename followed by a hash, followed by the function name:

```
maths.wmls#suar()
```


WML Scripts Standard Libraries:

There are six standard libraries totally. Here is an overview of them:

- **Lang:** The Lang library provides functions related to the WMLScript language core.
Example Function: abs(), abort(), characterSet(), float(), isFloat(), isInt(), max(), isMax(), min(), minInt(), maxInt(), parseFloat(), parseInt(), random(), seed()
- **Float:** The Float library contains functions that help us perform floating-point arithmetic operations.
Example Function: sqrt(), round(), pow(), ceil(), floor(), int(), maxFloat(), minFloat()
- **String:** The String library provides a number of functions that help us manipulate strings.
Example Function: length(), charAt(), find(), replace(), trim(), compare(), format(), isEmpty(), squeeze(), toString(), elementAt(), elements(), insertAt(), removeAt(), replaceAt()
- **URL:** The URL library contains functions that help us manipulate URLs.
Example Function: getPath(), getReferer(), getHost(), getBase(), escapeString(), isValid(), loadString(), resolve(), unescapeString(), getFragment()
- **WMLBrowser:** The WMLBrowser library provides a group of functions to control the WML browser or to get information from it.
Example Function: go(), prev(), next(), getCurrentCard(), refresh(), getVar(), setVar()
- **Dialogs:** The Dialogs library Contains the user interface functions.
Example Function: prompt(), confirm(), alert()

WML Scripts Comments:

There are two types of comments in WMLScript:

- **Single-line comment:** To add a single-line comment, begin a line of text with the // characters.
- **Multi-line comment:** To add a multi-line comment, enclose the text within /* and */.

These rules are the same in WMLScript, JavaScript, Java, and C++. The WMLScript engine will ignore all comments. The following WMLScript example demonstrates the use of comments:

```
// This is a single-line comment.
```

```
/* This is a
```

```
multi-line comment. */
```

```
/* A multi-line comment can be placed on a single line. */
```

WML Script Case Sensitivity:

The WMLScript language is case-sensitive. For example, a WMLScript function with the name WMLScript Function is different from wmlscript function. So, be careful of the capitalization when defining or referring to a function or a variable in WMLScript.

Whitespaces in WMLScript:

Except in string literals, WMLScript ignores extra whitespaces like spaces, tabs and newlines. Hence, the code in the earlier "Hello World" example can be typed in the following way and the result will remain the same:

WML Script Statement Termination by Semicolons:

A semicolon is required to end a statement in WMLScript. This is the same as C++ and Java. Note that JavaScript does not have such requirement but WML Script makes it mandatory.

WAP - Core Services

A vast majority of WAP services are available in the market. You may contact to some WAP lover to have a big list of all the available services and then you can start accessing those services from your WAP enabled mobile phone.

However, some examples of useful mobile services are in the following fields:

Banking:

- Accessing account statements.
- Paying bills.
- Transferring money between accounts.

Finance:

- Retrieving stock and share prices.
- Buying and selling stocks and shares.
- Looking up interest rates.
- Looking up currency exchange rates.

Shopping:

- Buying everyday commodities.
- Browsing and buying books.
- Buying CDs.

Ticketing:

- Booking or buying airline tickets.
- Buying concert tickets.
- Booking theatre tickets.

Entertainment:

- Retrieving restaurant details.
- Looking up clubs.
- Finding out what is playing in what cinemas.
- Playing solitaire games.
- Playing interactive games.

Weather:

- Retrieving local weather forecasts.
- Looking up weather at other locations.

E- Messaging:

- Voice mail.
- Unified Messaging.
- Enhanced support of legacy SMS services.

Live WAP Examples:

The following are some example WAP applications:

- **123Jump** (<http://www.123jump.com>) A selection of stock data and news, all via WAP.
- **1477.com** (<http://1477.com>) WAP/Web development services.
- **2PL World-Wide Hotel Guide** (<http://wap.2pl.com>) A worldwide hotel guide accessible in multiple languages via a WAP-enabled device.
- **AEGEE-Eindhoven** (<http://wappy.to/aegee/>) A Europe-wide students association, whose goal is to allow all students to integrate and learn about each others cultures.

- **Ajaxo (<http://www.ajaxo.com>)** A WAP service for Wireless Stock Trading from any WAP-enabled device.
- **Aktiesidan (<http://mmm.aktiesidan.com/servlets/aktiesidan/>)** A Swedish stock-market monitoring service, all WAP-enabled.
- **Amazon.com Bookshop (<http://www.amazon.com/phone/>)** Amazon.com has launched this WAP portal (HDML-based) for browsing books.
- **Traffic Maps (<http://www.webraska.com/>)** A French service that monitors and shows the latest in traffic news via maps.

WAP - Key Benefits

The following sections outline how various groups may gain from WAP:

Subscribers:

It is crucial that the subscribers will benefit from using WAP based services, otherwise, there will be no incentive neither for WAP as a whole nor for any of the other groups mentioned below. The key-benefits can be summarised as:

- Portability
- Easy to use
- Access to a wide variety of services on a competitive market
- The possibility of having personalised services
- Fast, convenient, and efficient access to services
- To fulfil as many customers needs as possible, WAP devices will be available in various form factors, e.g. pagers, handheld PCs, and phones

Operators:

Many of the advantages mentioned under "Service Providers" are be applicable to operators as well. The operator's benefits may include:

- Address new market segments of mobile users by enabling a wider range of mobile VAS.
- Deploy telephony services that in contrast to traditional telephony services are easy to create, update, and personalise
- Use the flexibility of WAP as a tool to differentiate from competitors
- Attractive interface to services will increase usage
- Increased revenues per user due to higher network utilisation

- Convenient service creation and maintenance including short time-to-market
- Replace expensive customer care centres with WAP based services (E-care)
- WAP services are designed to be independent of the network, implying that an operator who runs different types of networks only have to develop its services ones
- An open standard means that equipment will be provided by many manufacturers

Service Providers:

WAP opens new possibilities for service and content providers since they not necessarily have to come to an agreement with a specific operator about providing services to their customers. The gains are for example:

- Create a service once, make it accessible on a broad range of wireless networks
- Address new market segments by launching innovative mobile VAS. Keep old customers by adapting existing Internet services to WAP
- Keep old customers by adapting existing Internet services to WAP
- Convenient service creation and maintenance
- Creating a WAP service is no harder than creating an Internet service today since WML and WMLScript are based on well-known Internet technology
- Use standard tools like ASP or CGI to generate content dynamically
- Utilise existing investments in databases, etc that are the basis of existing Internet services

Manufacturers:

Mobile devices supporting WAP will be available in many different form factors, e.g., cellular phones, pagers, and handheld PCs. Hardware manufacturers will also need to supply operators etc with equipment, such as WAP Gateway/Proxys and WTA servers. Manufacturer benefits are for example:

- WAP scales across a broad range of mobile networks, meaning that WAP implementations can be used in devices supporting different types of networks.
- The expected wide adoption of WAP implies that economies of scales can be achieved, meaning that the huge mass-market can be addressed
- The fact that WAP is designed to consume minimal amount of memory, and that the use of proxy technology relieves the CPU, means that inexpensive components can be used in the handsets

- Reuse the deep knowledge about wireless network infrastructure to develop advanced servers that seamlessly integrates mobile VAS with telephony
- Seize the opportunity to introduce new innovative products

Tools Providers:

Today, there are large amount of tools available for creating applications for the web. Content developers have become used to the convenience that tools like FrontPage and DreamWeaver provides. Tools providers will be able to:

- Reuse and modify existing products to support WAP or even integrate WAP support in existing tools.
- Address a new customer base in the wireless community.

WAP - Modern Devices

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.

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 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.

WAP - Future Prospects

The future of WAP depends largely on whether consumers decide to use WAP devices to access the Web, and also on whether a new technology comes along that would require a different infrastructure than WAP.

On the consumer side, the factors largely involve the limitations of WAP and of handheld devices, the low bandwidth, the limited input ability, and the small screens all require users to adapt from their regular Web-browsing expectations.

In the next few years, mobile phones will start to benefit from very high bandwidth capabilities. 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 GPRS, EDGE, HSCSD, and UMTS:

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 bottom line is that WAP is not and can never be the Web on your mobile phone. WAP is great as long as developers understand that it's what's inside the applications that matters, and the perceived value of the content to the user. The browser interface itself, while important will always be secondary to the content.

WAP - Summary

In this tutorial, you have been introduced to all the basic concepts of WAP and WML, WMLScript, and the WAP architecture.

WAP key features benefits have also been discussed together with a look at what the future holds in this quickly moving market.

What is Next ?

Now you have basic understanding on WAP. The next step after WAP can be to learn any of the following technologies.

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. A main benefit is that users are always connected online, and will be charged only for the amount of data that is transported.

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.

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.

This new technology will deliver low-cost, high-capacity wireless communications, offering data rates of 1Mbps to 2Mbps with global roaming and other advanced UMTS services.

Now, if you need more detail about WAP technology then I would recommend you to go through other WAP resources listed in [WAP Useful Resources](#) chapter.