



W@P

wireless application protocol

tutorialspoint
SIMPLY EASY LEARNING

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

WAP is an international standard establishing how mobile devices can access information on the Internet.

It is a widely used set of protocols used on wireless devices such as mobile phones and PDAs. This tutorial helps you in understanding WAP technology starting from its basics.

Audience

This tutorial is designed for all those individuals who are looking for a starting point of learning WAP and we cover topics suited for both a total beginner and an advanced user.

Prerequisites

Before proceeding with this tutorial, it is advisable for you to understand the basics concepts of TELECOM.

Disclaimer & Copyright

© Copyright 2015 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute, or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness, or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial.....	i
Audience	i
Prerequisites	i
Disclaimer & Copyright.....	i
Table of Contents.....	ii
1. WAP – INTRODUCTION.....	1
Who is behind WAP?.....	1
Why is WAP Important?.....	2
WAP Microbrowser.....	2
2. WAP – KEY FEATURES.....	3
3. WAP – THE MODEL.....	4
The Internet Model	4
The WAP Model	5
How WAP Model Works?	6
4. WAP – ARCHITECTURE	7
Layers of WAP Protocol	7
5. WAP – ENVIRONMENT	9
Components of WAE	9
Hardware and Software Requirement.....	9
Configure Web Server for WAP	10
6. WAP – WML SYNTAX.....	11
WAP Program Structure	11
WML Commands.....	12

Sample WML Program.....	14
7. WAP – WML SCRIPT	15
WML Script Components.....	15
WML Script Operators.....	15
WAP - WML Operators.....	15
WML Script Control Statements	19
WAP - WML Control Statements	19
WMLScript Functions	21
WMLScript Standard Libraries.....	22
WML Scripts Comments	23
WML Script Case Sensitivity	23
Whitespaces in WMLScript.....	23
WMLScript Statement Termination by Semicolons	23
8. WAP – CORE SERVICES	24
Banking	24
Finance.....	24
Shopping	24
Ticketing.....	24
Entertainment.....	25
Weather	25
E- Messaging	25
Live WAP Examples	25
9. WAP – KEY BENEFITS	27
Subscribers.....	27
Operators.....	27

Service Providers.....	28
Manufacturers	28
Tools Providers.....	28
10. WAP – MODERN DEVICES.....	30
11. WAP – FUTURE PROSPECTS.....	31
12. WAP – SUMMARY.....	32
What is Next?.....	32

1. 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**. The dictionary definition of these terms are as follows:

- **Wireless:** Lacking or not requiring wires for 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 the 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 an Internet and a mobile phone was a mobile phone. You could surf the internet, do serious research, or be entertained on the Internet using your computer, but this was limited to the computer only.

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.

Because of being open and secure, WAP is well suited for many different applications including stock market information, weather forecasts, enterprise data, games, and many more.

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. Further, in the coming years, more development tools will be announced.

WAP Microbrowser

To browse a standard internet website, you need a web browser. Similarly, 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.

2. WAP – KEY FEATURES

Here, in this chapter, we have listed a few 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 as both the service developers and the 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 into the 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 possible to add a procedural logic and computational functions to WAP based 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.

3. WAP – THE MODEL

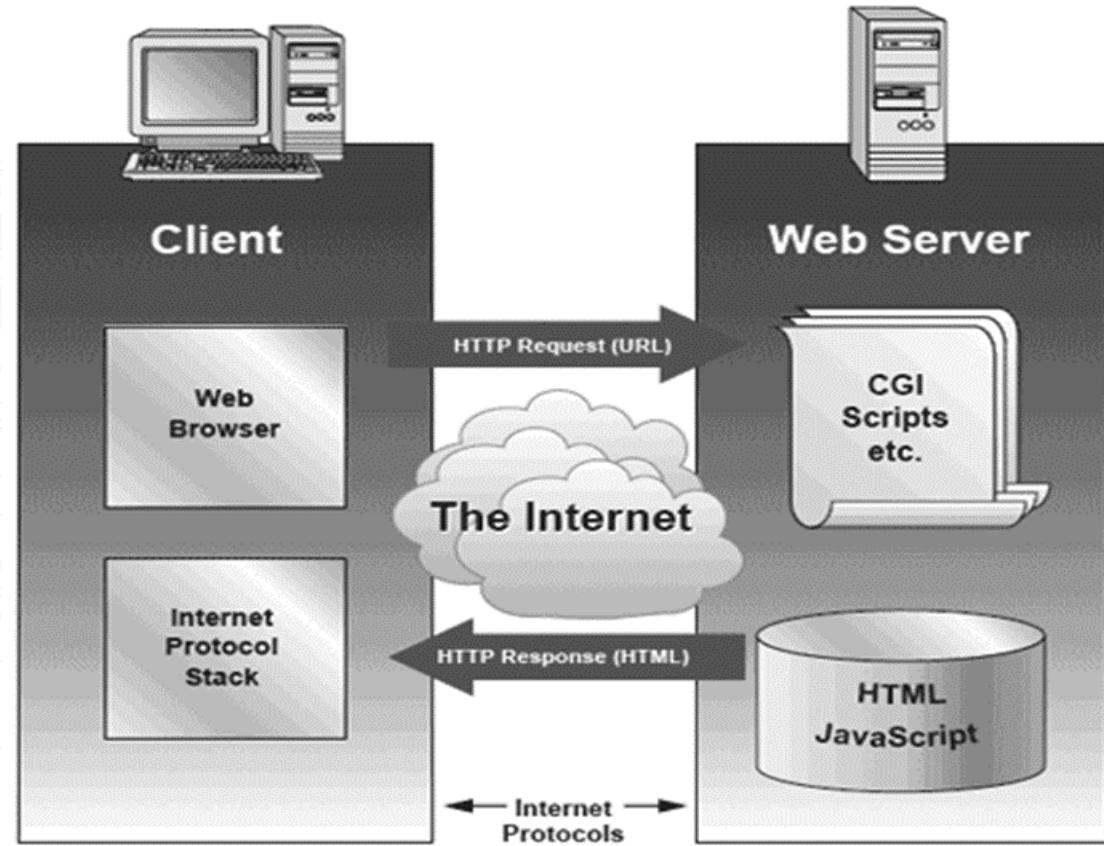
Before we describe WAP model, first we would like you to understand how a Standard Internet works.

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 utilized.

The figure below shows how a WWW client requests 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.

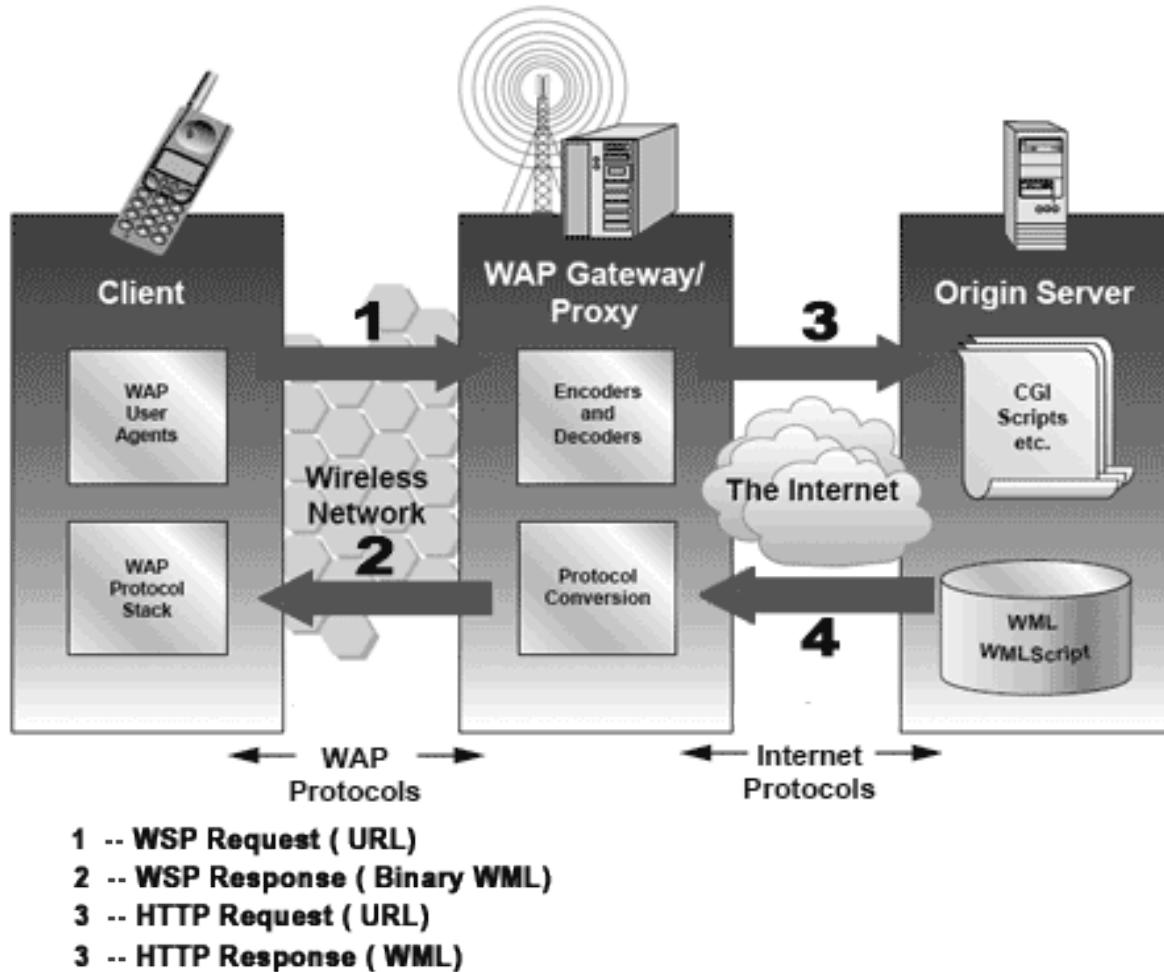


The content available at the web server may be static or dynamic. Static content is produced once and not changed or updated very often; for example, a company presentation. Dynamic content is needed when the information provided by the service provider changes more often; for example, timetables, news, 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. It is quite similar to the Internet model. Without the WAP Gateway/Proxy, the two models would have been practically identical.



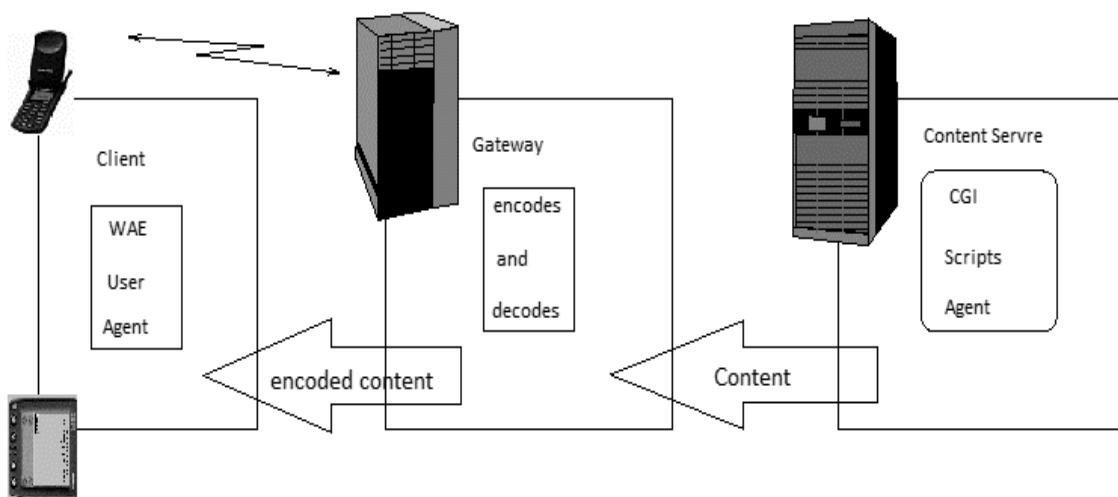
WAP Gateway/Proxy is the entity that connects the wireless domain with the Internet. You should make a note that the request that is sent from the wireless client to the WAP Gateway/Proxy uses the Wireless Session Protocol (WSP). In its essence, WSP is a binary version of HTTP.

A **markup language** - the Wireless Markup Language (WML) has been adapted to develop the 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 as follows:

- The users select an option on their mobile devices, which have 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 places 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 (as shown in the image given below).
- 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.



4. 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:

Layers of WAP Protocol

Application Layer

Wireless Application Environment (WAE). It is much popular among the content developers, as 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 the top of a datagram service, such as User Datagram Protocol (UDP). It is a part of the standard suite of TCP/IP protocols used to provide a simplified protocol that is suitable for low bandwidth wireless stations.

Security Layer

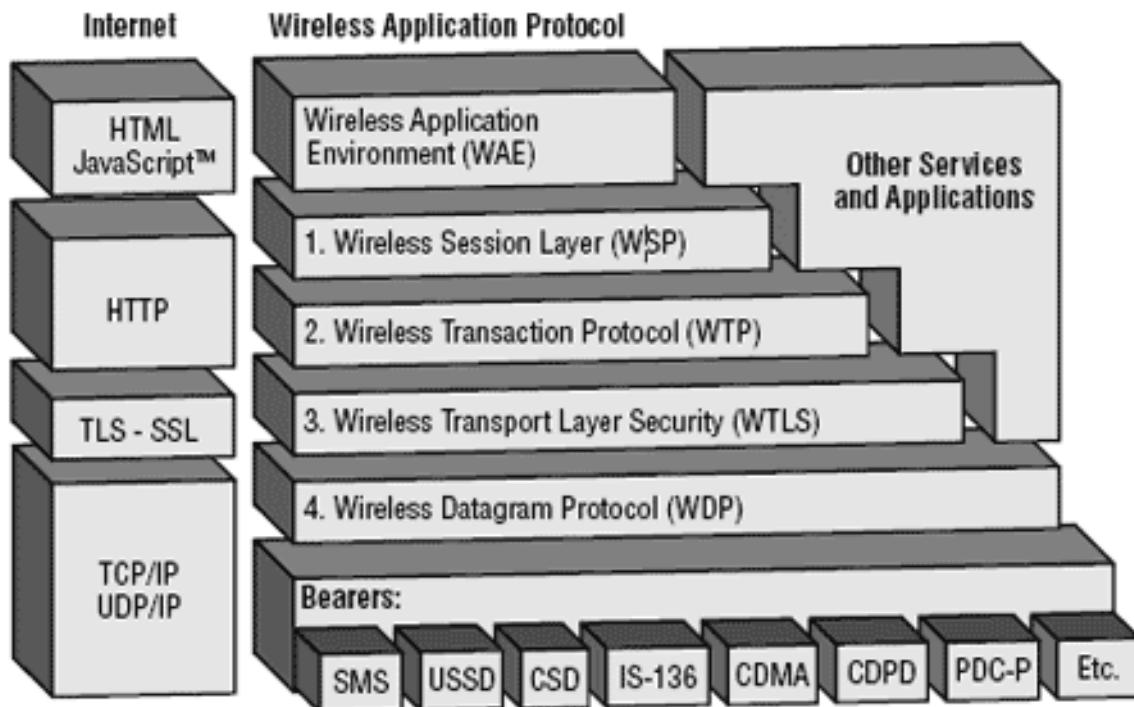
Wireless Transport Layer Security (WTLS). WTLS incorporates security features that are based upon the established Transport Layer Security (TLS) protocol standard. It includes data integrity checks, privacy, service denial, and authentication services.

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 utilize 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 shown above are not a part of the WAP protocol stack.

5. WAP – ENVIRONMENT

Wireless Application Environment (WAE), the uppermost layer in the WAP stack, provides an environment that enables a wide range of applications to be used on the wireless devices. We have earlier discussed about the WAP WAE programming model. In this chapter, we will focus on the various components of WAE.

Components of WAE

Addressing Model

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

Wireless Markup Language (WML)

WML is 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 the services, which have been written in WML in the way that it 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

The minimum requirement to develop a WAP application is a web server and a WAP simulator. Using the 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 a 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. 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/vmd.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.

6. 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 on 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 the 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
<code><p></code>	Paragraph
<code></code>	Bold
<code><big></code>	Large
<code></code>	Emphasized
<code><i></code>	Italicized
<code><small></code>	Small
<code></code>	Strongly Emphasized
<code><u></code>	Underlined
<code>
</code>	Line Break

Inserting Images

```

```

Using Tables

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

Variables

Declared as:

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

Used as:

```
$ identifier or
$ (identifier) or
$ (Identifier; conversion)
```

Forms

Command	Description
<select>	Defines 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>

```

7. 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 gives a brief description of all the important WML Script components.

WML Script Components

WML Script is very similar to Java Script. WML Script components have almost similar meaning as they have in Java Script. The WML Script program components are summarized here.

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**.

WAP - WML Operators

Arithmetic Operators

Following are the arithmetic operators supported by the WML Script language:

Assume variable 'A' holds 10 and variable 'B' holds 20 then:

Operator	Description	Example
+	Adds two operands	A + B will give 30
-	Subtracts second operand from the first	A - B will give -10
*	Multiply both operands	A * B will give 200
/	Divide numerator by denominator	B / A will give 2

%	Modulus Operator and remainder of after an integer division	B % A will give 0
++	Increment operator, increases integer value by one	A++ will give 11
--	Decrement operator, decreases integer value by one	A-- will give 9

Comparison Operators

Following are the comparison operators supported by the WML Script language:

Assume variable 'A' holds 10 and variable 'B' holds 20 then:

Operator	Description	Example
==	Checks if the value of two operands is equal or not, if yes, then condition becomes true.	(A == B) is not true.
!=	Checks if the value of two operands is equal or not, if values are not equal, then condition becomes true.	(A != B) is true.
>	Checks if the value of left operand is greater than the value of right operand, if yes, then condition becomes true.	(A > B) is not true.
<	Checks if the value of left operand is less than the value of right operand, if yes, then condition becomes true.	(A < B) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes, then condition becomes true.	(A >= B) is not true.
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then, condition becomes true.	(A <= B) is true.

Logical Operators

Following are the logical operators supported by the WML Script language:

Assume variable 'A' holds 10 and variable 'B' holds 20 then:

Operator	Description	Example
and	Called Logical AND operator. If both the operands are true, then condition becomes true.	(A and B) is true.
or	Called Logical OR Operator. If any of the two operands is non-zero, then condition becomes true.	(A or B) is true.
&&	Called Logical AND operator. If both the operands are non-zero, then condition becomes true.	(A && B) is true.
	Called Logical OR Operator. If any of the two operands is non-zero, then condition becomes true.	(A B) is true.
!	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true, then Logical NOT operator will make false.	!(A && B) is false.

Assignment Operators

Following are assignment operators supported by the WML Script language:

Operator	Description	Example
=	Simple assignment operator, Assigns values from right side operands to left side operand.	C = A + B will assign value of A + B into C
+=	AND assignment operator, It adds right operand to the left operand and assigns the result to left operand.	C += A is equivalent to C = C + A
-=	Subtract AND assignment operator. It subtracts right operand from the left operand and assigns the result to the left operand.	C -= A is equivalent to C = C - A
*=	Multiply AND assignment operator. It multiplies right operand with the left operand	C *= A is equivalent to C = C * A

	and assigns the result to left operand.	
/=	Divide AND assignment operator. It divides left operand with the right operand and assigns the result to left operand.	C /= A is equivalent to C = C / A
%=	Modulus AND assignment operator. It takes modulus using two operands and assigns the result to left operand.	C %= A is equivalent to C = C % A

Conditional Operator

There is one more operator called conditional operator. First, it evaluates an expression for a true or false value and then execute one of the two given statements depending upon the result of the evaluation. The conditional operator has the following syntax:

Operator	Description	Example
? :	Conditional Expression	If Condition is true? Then value X : Otherwise value Y

Operators Categories

All the operators that we have discussed above can be categorized into the following categories:

- Unary prefix operators, which precede a single operand.
- Binary operators, which take two operands and perform a variety of arithmetic and logical operations.
- The conditional operator (a ternary operator), which takes three operands and evaluates either the second or the third expression, depending on the evaluation of the first expression.
- Assignment operators, which assign a value to a variable.

Precedence of WML Script Operators

Operator precedence determines the grouping of terms in an expression. This affects how an expression is evaluated. Certain operators have higher precedence than others; for example, the multiplication operator has higher precedence than the addition operator:

For example, $x = 7 + 3 \times 2$; here, x is assigned 13, not 20 because operator \times has higher precedence than $+$ so it first get multiplied with 3×2 and then add 7.

Here, operators with the highest precedence appear at the top of the table, those with the lowest precedence appear at the bottom. Within an expression, higher precedence operators will be evaluated first.

Category	Operator	Associativity
Unary	! ++ --	Right to left
Multiplicative	* / %	Left to right
Additive	+ -	Left to right
Relational	< <= > >=	Left to right
Equality	== !=	Left to right
Logical AND	&&	Left to right
Logical OR		Left to right
Conditional	?:	Right to left
Assignment	= += -= *= /= %=	Right to left

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**.

WAP - WML Control Statements

WML Script **if...else** Statement

WMLScript's **if...else** statement uses the following syntax. The part inside brackets [] is optional. The syntax is the same as that of C++, Java, and JavaScript.

```

if (condition)
{
    WMLScript statement(s)
}
[else
{
    WMLScript statement(s)
}]

```

If condition is the Boolean value true, the statement(s) enclosed in the first curly brackets {} will be executed; if condition is false or invalid, the statement(s) enclosed in the second curly brackets {} will be executed.

WMLScript *while* Statement

WMLScript's **while** statement is used to repeat the execution of a block of statements while a condition is true. It has the following syntax:

```

while (condition)
{
    WMLScript statement(s)
}

```

The statement(s) enclosed in the curly brackets {} will be executed again and again as long as condition is true. The loop stops when condition evaluates to false or invalid.

WMLScript *for* Statement

Like a **while** loop, a **for** loop is executed repeatedly as long as a condition is satisfied. If the number of times to be repeated is certain, using the 'for' statement will be more convenient than using the 'while' statement. The 'for' statement has the following syntax in WMLScript. The parts enclosed in brackets [] are optional.

```

for ([expression1]; [expression2]; [expression3])
{
    WMLScript statement(s)
}

```

expression1 is the initialization expression. It is executed before any WMLScript statements in the 'for' loop are executed. expression1 is executed only once in the lifetime of the 'for' statement. expression2 is the conditional expression that determines whether the 'for loop' should continue or stop. expression3 is executed after each iteration.

WMLScript *break* Statement

The 'break' statement is used to quit a loop. It must be put inside the 'while loops' or the 'for loops.' The following WMLScript example demonstrates how to use the 'break statement:'

```
var result = 0;
for (var counter=0; counter<10; counter++)
{
    break;
    result += 5;
}
```

After the execution the above WMLScript code, the value of result is 0. This is because the break statement exits the 'for loop.' The statement "result += 5;" is never executed.

WMLScript *continue* Statement

The 'continue statement' is used to quit the current iteration of a loop in WMLScript. The next iteration will be started if the loop's conditional expression evaluates to true. The 'continue statement' must be put inside the 'while loops' or 'for loops.' The following script demonstrates how to use the 'continue statement:'

```
var result1 = 0;
var result2 = 0;
for (var counter=0; counter<10; counter++)
{
    result1 += 5;
    continue;
    result2 += 5;
}
```

After the execution of the above WMLScript code, the value of result1 is 20 and that of result2 is 0. This is because when the WMLScript interpreter encounters the continue statement, then it ends the current iteration. Hence, the statement "result2 += 5;" is never executed.

WMLScript 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 and then followed by the function name:

```
maths.wmls#squar()
```

WMLScript Standard Libraries

The six standard libraries of WMLScript are described below:

- **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 to 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 to 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 to 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 a 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.

WMLScript 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 WMLScript makes it mandatory.

8. 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 to different 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 updates

E-Messaging

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

Live WAP Examples

Following are some of the examples of 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 can be accessed in multiple languages via a WAP-enabled device.
- **AEGEE-Eindhoven (<http://wappy.to/aegee/>)**: A Europe-wide students association, whose objective is to facilitate all students to integrate and learn about each other's 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 traffic news via maps.

9. WAP – KEY BENEFITS

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

Subscribers

It is crucial that the subscribers benefit from using WAP based services. The key benefits can be summarized as:

- Portability
- Easy to use
- Access to a wide variety of services on a competitive market
- The possibility of having personalized 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 personalize.
- 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 utilization.
- Convenient service creation and maintenance including short time-to-market.
- Replace expensive customer care centers 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 has to develop these services only once.
- 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. For example, the gains are:

- 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.
- Convenient service creation and maintenance.
- Creating a WAP service is not harder than creating an Internet service today because WML and WMLScript are based on well-known Internet technology.
- Use standard tools like ASP or CGI to generate content dynamically.
- Utilize 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 with equipment such as WAP Gateway/Proxys and WTA servers. Manufacturers benefit in the following ways:

- 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, we have a number of tools available for creating applications for the web. Content developers have become accustomed 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.

10. WAP – MODERN DEVICES

A WAP device is a combination of hardware and software capable of running a WAP-compliant micro-browser, 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 as a web browser runs any HTML application. Once you have a WAP phone, you can access the Internet simply by entering URLs and following the links that appear.

These devices provide easy and secure access to the internet content and other services such as banking, leisure, and unified messaging. Furthermore, access is not restricted only to the Internet, but it can also deal with intranet information in the same way as internet content because both are based upon HTML.

Following is a selection of WAP phones that have been announced recently:



11. 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.

12. WAP – SUMMARY

In this tutorial, you have been introduced to all the basic concepts of WAP, 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 384 Kbps, or twice that available from GPRS alone.

It has been 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)

HSCSD is a new high-speed implementation of GSM data techniques. It uses four radio channels simultaneously. It is capable enough to allow users to access the Internet via the GSM network at 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.

Please send me your feedback and suggestion at webmaster@tutorialspoint.com