WML

wireless markup language

tutorialspoint

SIMPLYEASYLEARNING

www.tutorialspoint.com

https://www.facebook.com/tutorialspointindia

https://twitter.com/tutorialspoint

About the Tutorial

WML is an XML language used to specify content and user interface for WAP devices like PDA and Mobile Phones. The WAP forum provides a DTD for WML.

This tutorial explains how to use WML to develop WAP applications.

Audience

This tutorial is designed for Software Professionals who are in the need of learning the basics of WML.

Prerequisites

Before proceeding with this tutorial, you should have a basic understanding of XML, text editor, execution of programs, etc.

Disclaimer & Copyright

Table of Contents

1. WML – OVERVIEW

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

- WML stands for **W**ireless **M**arkup **L**anguage
- WML is an application of XML, which is defined in a document-type definition.
- WML is based on HDML and is modified so that it can be compared with HTML.
- WML takes care of the small screen and the low bandwidth of transmission.
- WML is the markup language defined in the WAP specification.
- WAP sites are written in WML, while web sites are written in HTML.
- WML is very similar to HTML. Both of them use tags and are written in plain text format.
- WML files have the extension ".wml". The MIME type of WML is "text/vnd.wap.wml".
- WML supports client-side scripting. The scripting language supported is called WML Script.

WML Versions

WAP Forum has released a latest version WAP 2.0. The markup language defined in WAP 2.0 is XHTML Mobile Profile (MP). The WML MP is a subset of the XHTML. A style sheet called **WCSS** (WAP CSS) has been introduced along with XHTML MP. The WCSS is a subset of the CSS2.

Most of the new mobile phone models released are WAP 2.0-enabled. Because WAP 2.0 is backward compatible to WAP 1.x, WAP 2.0-enabled mobile devices can display both XHTML MP and WML documents.

WML 1.x is an old technology. However, that does not mean it is of no use, since a lot of wireless devices that support only WML 1.x are still being used. The latest version of WML is 2.0 and it is backward compatible. So, WAP site developers need not worry about WML 2.0.

WML Decks and Cards

The main difference between HTML and WML is that the basic unit of navigation in HTML is a page, while that in WML is a card. A WML file can contain multiple cards and they form a deck.

When a WML page is accessed from a mobile phone, all the cards in the page are downloaded from the WAP server. So, if the user goes to another card of the same deck, the mobile browser does not have to send any requests to the server since the file that contains the deck is already stored in the wireless device.

You can put links, text, images, input fields, option boxes, and many other elements in a card.

WML Program Structure

Following is the basic structure of a WML program:

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

<wml>

<card id="one" title="First Card">
<p>
This is the first card in the deck
</p>
</card>

<card id="two" title="Second Card">
<p>
Ths is the second card in the deck
</p>
</card>

</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).

One WML deck (i.e. page) can have one or more cards as shown above. We will see complete details on WML document structure in subsequent chapter.

Unlike HTML 4.01 Transitional, text cannot be enclosed directly in the <card>...</card> tag pair. So, you need to put a content inside ... as shown above.

WAP Site Design Considerations

Wireless devices are limited by the size of their displays and keypads. It's therefore very important to take this into account when designing a WAP Site.

While designing a WAP site, you must ensure that you keep things simple and easy to use. You should always keep in mind that there are no standard micro browser behaviors and that the data link may be relatively slow, at around 10Kbps. However, with GPRS, EDGE, and UMTS, this may not be the case for long, depending on where you are located.

The following are general design tips that you should keep in mind when designing a service:

- Keep the WML decks and images to less than 1.5KB.
- Keep the text brief and meaningful, and as far as possible try to pre-code options to minimize the rather painful experience of user data entry.
- Keep the URLs **brief** and easy to recall.
- Minimize menu levels to prevent users from getting lost and the system from slowing down.
- Use standard layout tags such as <big> and <b>, and logically structure your information.
- Don't go overboard with the use of graphics, as many target devices may not support them.

2. WML – ENVIRONMENT

To develop WAP applications, you will need the following:

- **A WAP enabled Web Server:** You can enable your Apache or Microsoft IIS to serve all the WAP client request.
- **A WAP Gateway Simulator:** This is required to interact to your WAP server.
- **A WAP Phone Simulator:** This is required to test your WAP Pages and to show all the WAP pages.

You can write your WAP pages using the following languages:

- Wireless Markup Language (WML) to develop WAP application.
- WML Script to enhance the functionality of WAP application.

Configuring Web Server

In normal web applications, 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.

To make your Apache WAP compatible, you don't have to put a lot of effort. All that you need to do is to add support for the MIME types and extensions listed below.

File Extension	MIME type
WML (.wml)	text/vnd.wap.wml
WMLScript (.wmls)	text/vmd.wap.wmlscript
WMLScriptc (.wmlsx)	application/vnd.wap.wmlscriptc
WMLC (.wmlc)	application/vnd.wap.wmlc
WBMP (.wbmp)	image/vnd.wap.wbmp

Configure Apache Web Server for WAP

Let us assume you have Apache Web server installed on your machine. Now we will explain let you know tell you how to enable WAP functionality in your Apache web server.

Locate Apache's file httpd.conf which is usually in /etc/httpd/conf, and add the following lines to the file and restart the server:

```
AddType text/vnd.wap.wml .wml

AddType text/vnd.wap.wmlscript .wmls
```

```
AddType application/vnd.wap.wmlc .wmlc
AddType application/vnd.wap.wmlscriptc .wmlsc
AddType image/vnd.wap.wbmp .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.

Configure Microsoft IIS for WAP

To configure a Microsoft IIS server to deliver WAP content, you need to perform the following exercise:

- Open the Internet Service Manager console and expand the tree to view your Website entry. You can add the WAP MIME types to the whole server or to the individual directories.

- Open the Properties dialog box by right-clicking the appropriate server or directory, then choose Properties from the menu.

- From the Properties dialog, choose the HTTP Headers tab, then select the File Types button at the bottom right.

- For each MIME type listed in the above table, supply the extension with or without the dot (it will be automatically added for you), then click OK in the Properties dialog box to accept your changes.

Installing WAP Gateway Simulator

There are many WAP Gateway Simulator available on the Internet, so download any of them and install on your PC. You would need to run this gateway before starting WAP Mobile simulator.

WAP Gateway will take your request and passes it to the Web Server and whatever response will be received from the Web server that will be passed to the Mobile Simulator.

You can download it from Nokia website:

- **Nokia WAP Gateway simulator** - Download Nokia WAP Gateway simulator.

Installing WAP Phone Simulator

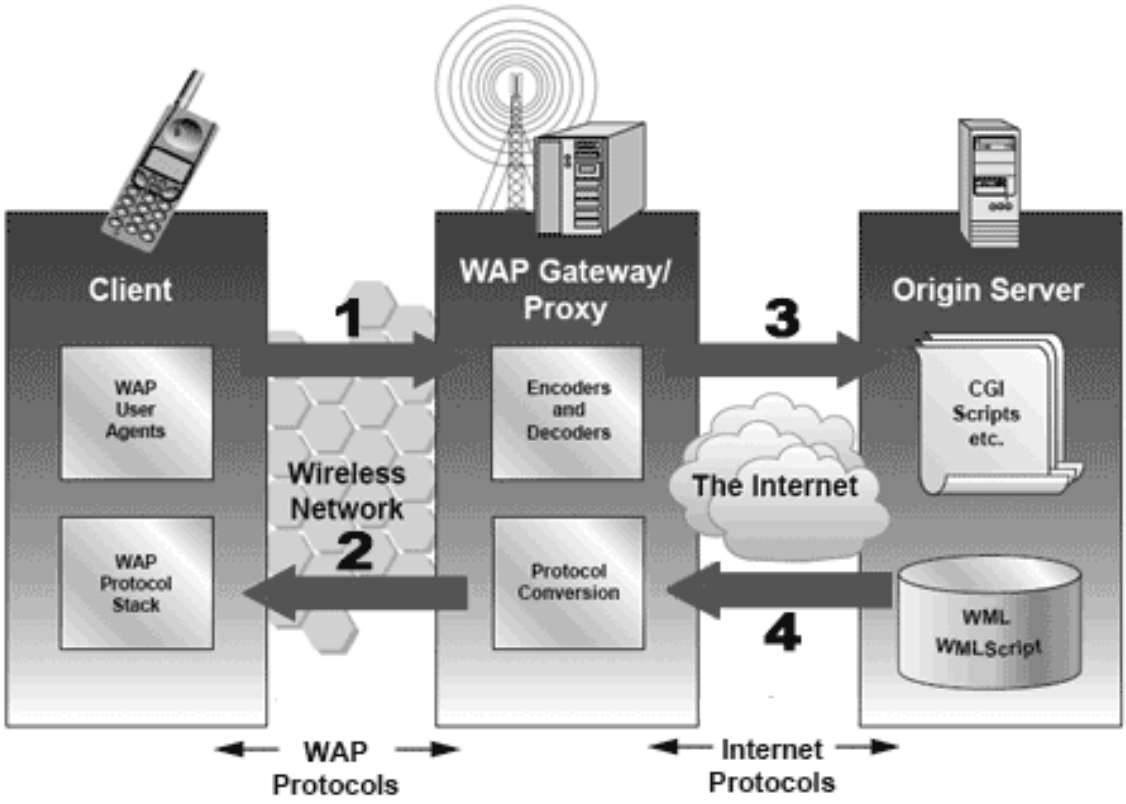
There are many WAP Simulators available on the Internet, so download any of them and install on your PC, which you will use as a WAP client. Here are popular links to download simulator:

- **Nokia WAP simulator** - Download Nokia WAP simulator.
- **WinWAP simulator** - Download WinWAP browser from their official website.

NOTE: If you have WAP enabled phone, then you do not need to install this simulator. But while doing development, it is more convenient and economic to use a simulator.

The WAP Model

The following figure 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, which 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, which 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.

3. WML – SYNTAX

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

Following is the basic structure of a WML program:

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

<wml>

<card id="one" title="First Card">
<p>
This is the first card in the deck

</p>
</card>

<card id="two" title="Second Card">
<p>
Ths is the second card in the deck
</p>
</card>

</wml>
```

WML Document Prolog

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). The DTD referenced is defined in WAP 1.2, 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 prolog components are not WML elements and they should not be closed, i.e. you should not give them an end tag or finish them with />.

WML Document Body

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

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

Unlike HTML 4.01 Transitional, text cannot be enclosed directly in the <card>...</card> tag pair. So you need to put a content inside ... as shown above.

Testing Your Program

Put the above code in a file called test.wml file, and put this WML file locally on your hard disk, then view it using an emulator.

This is by far the most efficient way of developing and testing WML files. Since your aim is, however, to develop a service that is going to be available to WAP phone users, you should upload your WML files onto a server once you have developed them locally and test them over a real Internet connection. As you start developing more complex WAP services, this is how you will identify and rectify performance problems, which could, if left alone, lose your site visitors.

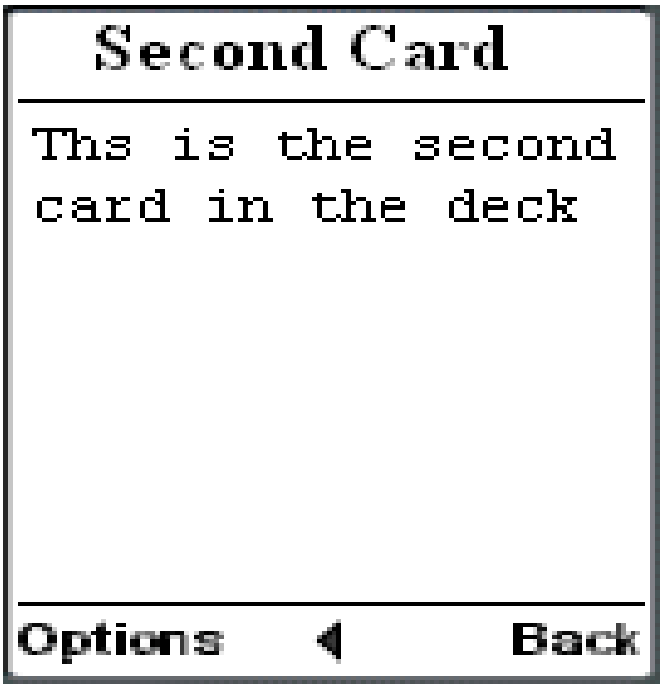
In uploading the file test.wml to a server, you will be testing your WML emulator to see how it looks and behaves, and checking your Web server to see that it is set up correctly. Now start your emulator and use it to access the URL of test.wml. For example, the URL might look something like this:

```
http://websitename.com/wapstuff/test.wml
```

NOTE: Before accessing any URL, make sure WAP Gateway Simulator is running on your PC.

When you will download your WAP program, then you will see only first card at your mobile. Following is the output of the above example on Nokia Mobile Browser 4.0. This mobile supports horizontal scrolling. You can see the text off the screen by pressing the "Left" or "Right" button.

When you press right button, then second card will be visible as follows:



4. WML – ELEMENTS

WML is defined by a set of *elements* that specify all markup and structural information for a WML deck. Elements are identified by tags, which are each enclosed in a pair of angle brackets.

Unlike HTML, WML strictly adheres to the XML hierarchical structure, and thus, elements must contain a start tag; any content such as text and/or other elements; and an end tag. Elements have one of the following two structures:

- **<tag> content </tag>** : This form is identical to HTML.
- **<tag />:** This is used when an element cannot contain visible content or is empty, such as a line break. WML document's prolog part does not have any element, which has closing element.

Following table lists the majority of valid elements. A complete detail of all these elements is given in **WML Tags Reference**.

Deck & Card Elements

WML Elements	Purpose
<!-->	Defines a WML comment
<wml>	Defines a WML deck (WML root)
<head>	Defines head information
<meta>	Defines meta information
<card>	Defines a card in a deck
<access>	Defines information about the access control of a deck
<template>	Defines a code template for all the cards in a deck

Text Elements

WML Elements	Purpose
** **	Defines a line break
<p>	Defines a paragraph
<table>	Defines a table
<td>	Defines a table cell (table data)
<tr>	Defines a table row
<pre>	Defines preformatted text

Text Formatting Tags

WML Elements	Purpose
<b>	Defines bold text
<big>	Defines big text
****	Defines emphasized text
<i>	Defines italic text
<small>	Defines small text

****	Defines strong text
<u>	Defines underlined text

Image Elements

WML Elements	Purpose
<img>	Defines an image

Anchor Elements

WML Elements	Purpose
<a>	Defines an anchor
<anchor>	Defines an anchor

Event Elements

WML Elements	Purpose
<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

Task Elements

WML Elements	Purpose
<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.

Input Elements

WML Elements	Purpose
<input>	Defines an input field
<select>	Defines a select group
<option>	Defines an option in a selectable list
<fieldset>	Defines a set of input fields
<optgroup>	Defines an option group in a selectable list

Variable Elements

WML Elements	Purpose
<setvar>	Defines and sets a variable
<timer>	Defines a timer

5. WML – COMMENTS

As with most programming languages, WML also provides a means of placing comment text within the code.

Comments are used by developers as a means of documenting programming decisions within the code to allow for easier code maintenance.

WML comments use the same format as HTML comments and use the following syntax:

```
<!-- This will be assumed as a comment -->
```

A multiline comment can be given as follows:

```
<!-- This is a multi-line
     comment -->
```

The WML author can use comments anywhere, and they are not displayed to the user by the user agent. Some emulators may complain if comments are placed before the XML prolog.

Note that comments are not compiled or sent to the user agent, and thus have no effect on the size of the compiled deck.

6. WML – VARIABLES

Because multiple cards can be contained within one deck, some mechanism needs to be in place to hold data as the user traverses from card to card. This mechanism is provided via WML variables.

WML is case sensitive. No case folding is performed when parsing a WML deck. All enumerated attribute values are case sensitive. For example, the following attribute values are all different: id="Card1", id="card1", and id="CARD1".

Variables can be created and set using several different methods. Following are the two examples:

The <setvar> element

The <setvar> element is used as a result of the user executing some task. The >setvar> element can be used to set a variable's state within the following elements: <go>, <prev>, and <refresh>.

This element supports the following attributes:

Attribute	Value	Description
name	string	Sets the name of the variable
value	string	Sets the value of the variable
class	class data	Sets the class name for the element.
id	element ID	A unique ID for the element.

The following element would create a variable named '*a*' with a value of 1000:

```
<setvar name="a" value="1000"/>
```

The input elements

Variables are also set through any input element like *input, select, option,* etc. A variable is automatically created that corresponds with the named attribute of an input element.

For example, the following element would create a variable named *b*:

```
<select name="b">
<option value="value1">Option 1</option>
<option value="value2">Option 2</option>
</select>
```

Using Variables

Variable expansion occurs at runtime, in the micro browser or emulator. This means, it can be concatenated with or embedded in other text.

Variables are referenced with a preceding dollar sign, and any single dollar sign in your WML deck is interpreted as a variable reference.

```
<p> Selected o
```

End of ebook preview

If you liked what you saw…

Buy it from our store @ https://store.tutorialspoint.com