

# Wireless Markup Language

## Introduction

The Wireless Markup Language (WML) is the HTML of WAP browsers and is transmitted via the HTTP protocol.

WML is a markup language built specifically for communicating across WAP-based networks, and is based upon XML (eXtensible Markup Language). Like HDML, it is at first glance similar to HTML, but is also a much more strictly written language.

To make it possible for web pages to be read from a WAP-enabled device, WML must be used. The WML coder determines within the code what parts of the web page are viewable to the device, and what is not. For example, it would not be too advantageous for a 468x60 pixel banner to be loaded into the small screen of a WAP device, due to size, color and bandwidth restraints. However, certain parts of the text may be made available to the device.

The advantage of the WML language is the fact that, since it is a subset of XML, developer's can easily kill two birds with one stone by building both the web page and wireless device page simultaneously. While this is still possible with HDML code, it is certainly not as obvious and workarounds must be introduced

There are a number of WML-specific sites and guides available on the web. Links to many of the better resources can be found on the next page of this article.

## History of WAP and WML

The roots of WAP are rather interesting, as they are built on the premise of industry cooperation. This is rather ironic, since the first official release of the protocol took place at the height of the ill-remembered 'Browser Wars'. It is undoubtedly this cooperation that was one of the major propellants for the widespread acceptance of WAP, allowing the standard to be quickly developed and integrated into the existing products of the many corporations responsible for its' development. This open standard also led to the rise of many new startups focused upon developing and marketing their own niche applications. Let's turn towards a brief discussion of this history, and how it ultimately led to the popularity seen today.

Way back when, in 1995, Ericsson spearheaded an effort to develop a general protocol that would offer a variety of value added services to wireless networks. Several other companies were soon on Ericsson's heels, developing various other technologies to compete in this soon-to-explode market, two major players including Nokia (<http://www.nokia.com>) and Phone.com, formally known as Unwired Planet (<http://www.phone.com>).

## WAP

The WAP protocol is the leading standard for information services on wireless terminals like digital mobile phones. WML is used to create pages that can be displayed in a WAP browser. In our WAP tutorial you will learn about WAP and WML, and how to convert your HTML pages to pocket format, so that your information can be accessed from devices like mobile WAP Phones.

In short, WAP provides a method to communicate across wireless networks quickly, securely and efficiently. Communication can take place using, but is not limited to, devices such as cell-phones, pagers, two-way radios and personal data assistants. This communication is not limited to static, boring pages like the web once was; Rather, WAP offers the opportunity to integrate databases, dynamic content, e-commerce, and secure information trafficking via a WAP-enabled device. Although the name itself refers to a single protocol, WAP can actually be thought of a compilation of protocols brought together to cover many aspect of wireless communications.

## **WML, XML, HDML AND HTML**

WML - the Wireless Markup Language - is the language now used to write content for WAP phones. The alternative, HDML (Handheld Device Markup Language), can be taken to be obsolete.

WML is a 'markup' language. This means that WML pages are written and saved as text files, using 'tags' like those found in HTML. Anyone familiar with HTML should find learning WML fairly easy

There are, however, some important general differences between HTML and WML. These stem from the fact that WML is specified in XML (the eXtensible Markup Language). XML specifies a general way to define different types of markup languages, and has the following quirks

XML-specified languages are case sensitive. So, for instance, a tag <wml> is different from a tag <WML>. This is unlike HTML, where case is generally ignored.

XML-specified languages are strict. Most HTML editors are very forgiving about badly coded HTML pages, and do their best to show some content. WML editors, however, will just report an error if given a badly coded WML page

All the attributes of elements must be contained within either double (") or single (') quotes. This is slightly different from HTML where attributes need not be contained within quotation marks for most browsers

One way in which WML is strict is that all tags crave closure. For instance, an opening <wml> tag must have a closing </wml> tag. Some tags don't come in pairs and are allowed to close themselves; these must have a forward slash at the end of their text, like <br/>.

## **Why Should we use WML**

Although you might not have any plans immediately for creating a WAP version of your site, it is always a good idea to get involved in new technology. All you need to do is make a small site (even one page) which tells people a bit about your website. In the future you can develop the site further with things like e-mail and information for people to get directly off their phones.

Over the past few months new WAP (Wireless Applications Protocol) phones have become extremely popular and many large websites have created special 'mobile' versions of their site. Many people predict that, over the next few years, WAP sites will become extremely popular and e-commerce over mobile phones will be widely available

The main sites which will benefit from WAP are ones providing a service like e-mail, live sports scores or a calendar service etc. but there are many other uses. For example, a site giving music reviews could put their reviews on a WAP site. People could then read the reviews on their mobile phone while browsing through the CDs in a shop.

## WML TAGS

WAP homepages are not very different from HTML homepages. The markup language used for WAP is WML (Wireless Markup Language). WML uses tags - just like HTML - but the syntax is stricter and conforms to the XML 1.0 standard.

WML pages have the extension \*.WML, just like HTML pages have the extension \*.HTML

WML is mostly about text. Tags that would slow down the communication with handheld devices are not a part of the WML standard. The use of tables and images is strongly restricted

Since WML is an XML application, all tags are case sensitive (<wml> is not the same as </WML>), and all tags must be properly closed.

## WML Decks and Cards

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. Navigation between the cards is done by the phone computer - inside the phone - without any extra access trips to the server.

```
<?xml version="1.0"?>
!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.techiwarehouse.com/DTD/wml_1.1.xml">
<wml>
<card id="HTML" title="HTML Tutorial">
< p> You can learn WML in 30 Days
</p>
</card>
< card id="XML" title="XML Tutorial">
< p> You can also learn how to make an WAP based Page
< /p>
</card>
```

```
</wml>
```

As you can see from the example, the WML document is an XML document. The DOCTYPE is defined to be wml, and the DTD is accessed at [www.techwarehouse.org/DTD/wml\\_1.1.xml](http://www.techwarehouse.org/DTD/wml_1.1.xml).

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAP/DTD WML 1.1//EN"
"http://www.wap.org/DTD/wml_1.1.xml">
<wml>
<card id="no1" title="Card 1">
<p>Hello World!</p>
</card>
<card id="no2" title="Card 2">
<p>Welcome to our WAP Tutorial!</p>
</card>
</wml>
```

## ELEMENTS

As WML is a language derived from XML, each element has an opening and a closing tag; if no actual content exists, an abbreviation can be used.

WML is therefore *case sensitive*, tags have to be given in lowercase letters.

```
<tag>Content</tag>
<tag/>
<tag attribute="value1">Content</tag>
<tag attribute="value"/>
```

## DECK

A *Deck* is best compared to a traditional HTML page and consists of one or more *Cards*. Always the complete deck is loaded to the device, therefore the deck should be small and contain logically pertaining cards.

A Deck has a <wml> Tag at the beginning (and a reciprocal closing </wml> at the end), and contains Meta-Tags and if needed <do>-elements good for all cards of this deck

## Prolog

Every valid WML document must contain a prolog giving the used XML version and WML Document Type Definition (DTD, in this case version 1.2).

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml_1.2.xml">
```

## Meta Tags

Proliferate information to search engines and are surrounded by a <head> tag. These are located at the head of the file after the prolog and the <wml> tag.

```
<head>
<meta name="keywords" content="keyword"/>
<meta name="description" content="a description"/>
<meta name="author" content="Thomas Ziegler"/>
</head>
```

## Comments

Comments are not displayed in the browser.

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

## Cards

A *Card* is the part of a Deck where the user navigates. He jumps from one <card> to another - which may reside in the same deck (in the same file) or somewhere else.

```
<card id="cardid" title="Card Title">
<p>
Text on this card
</p>
</card>
```

id: card id, used as jump target

title: A card title

## Links

Links may be given in two ways, but the <a> tag is preferred.

```
<anchor>Go to URL<go href="url.wml"/></anchor>
<anchor>Go to URL<go href="http://wml.org.de/url.wml"/></anchor>
<a accesskey="1" href="#card">Go to card</a>
<a href="http://wml.org/deck.wml/#card">Go to card</a>
```

href: Target; either a deck or a card; a card is prefixed by #, the target can either be given relative or absolute. If no card is given, the first card of a deck is linked. This address is called URL (*Uniform Resource Locator*).

## Attributes

Many elements (<a>, <input>, ...) can be attributed with positioning aids (for the order in which a Tab key press traverses them or for a shortcut key directly selecting an element).

```
<a tabindex="1" accesskey="1" href="#card1"/>
```

tabindex: order of selection with tab key

accesskey: key shortcut

## Masking

As with HTML special characters have to be masked to be displayed correctly.

&quot; Quotation Mark

&apos; Apostrophe

&amp; Ampersand

&lt; less than

&gt; greater than

&nbsp; non-breaking space

&shy; "soft hyphen", a conditional hyphenation

## Formating

```
<em>emphasized</em>
```

```
<strong>emphasized</strong>
```

```
<b>bold</b>
```

```
<i>italic</i>
```

```
<u>underlined</u>
```

```
<big>big</big>
```

```
<small>small</small>
```

## Paragraphs

```
<p mode="nowrap" align="right">
```

Example text aligned right

```
</p>
```

mode: "wrap", "nowrap" (break long lines)

align: "left", "center", "right" (alignment)

## Line Breaks

```
<br/>
```

## Tables

```
<table title="Title" columns="3">
```

```
<tr>
```

```

        <td>Line 1 Column 1</td>
        <td>Line 1 Column 2</td>
        <td>Line 1 Column 3</td>
    </tr>
    <tr>
        <td>Line 2 Column 1</td>
        <td>Line 2 Column 2</td>
        <td>Line 2 Column 3</td>
    </tr>
</table>

```

title: Title of the table

columns: Number of columns (mandatory)

tr: begins a new line (*Table Row*)

td: begins a new column (*Table Delimiter*)

## CDATA

With a CDATA section text can be displayed without interpretation of tags and special characters.

```

<![CDATA[Dies ist ein langer, unformatierter Text, der <nicht> geparkt wird,
daher sind auch "$%&" Zeichen erlaubt]]>

```

## IMAGES

The WBMP format (Wireless Bitmap) is used to code images under WML. Nokia has a DOS based conversion utility.

```



```

src: URL des Bildes

alt: Alternative text (mandatory)

align: "top", "middle", "bottom" (alignment)

## Select Boxes

Similar to dropdown fields in an HTML form there are WML select boxes.

```

<select>
<option onpick="link1.wml">Selection1</option>
<option onpick="link2.wml">Selection2</option>
<option onpick="link3.wml">Selection3</option>
</select>

```

onpick: Link to be followed on selection of a choice

## INPUT

`<input>` generates a text entry field and validates the entry against a format specification. The value is returned in a Variable.

```
<input name="Name" value="Template"/>
```

name: Name of the receiving Variable

value: Template

type: "text", "password"

format: Mask (e.g. A uppercase letters, N numbers)

maxlength: Maximum length of entry field

title: Title

A `<fieldset>` groups input tags logically.

## Tasks

Tasks trigger an action caused by an Event (Card entered, Link selected).

## GO

Jumps to a new location, either a Card or a Deck.

```
<go href="http://www.domain.de/deck"/>  
<go href="#card"/>
```

href: Target; a Cards is prefixed by #

## PREV

Jump to the previous location.

```
<prev/>
```

## REFRESH

Reload the current page.

```
<refresh/>
```

## NOOP

no operation - do nothing.

```
<noop/>
```



# DO

<do> handles Events.

<do> elements on Card level overwrite <do> tags on deck level with the same name, with <noop> a deck level event handler can be disabled locally.

On deck level <do> has to be surrounded by <template> and </template>.

The display is highly browser specific.

```
<do type="options" label="Label" name="Name">
  <go href="#card"/>
</do>
```

label: Describes the element

type: "accept", "prev", "options"

name: Name of the event handl

## Variables

Variables are a powerful enhancement of HTML functionality. They are used to proliferate status information or user entries from one Card to another or to the server.

Variables are prefixed by a dollar sign (\$). Therefore if a \$ sign should be used, you have to write \$\$.

```
<p>Value of T is $T</p>
```

Displays the text and the value of variable T.

Variables can be set in different ways:

```
<go href="#card" method="post">
  <postfield name="x" value="1"/>
  <setvar name="y" value="2"/>
</go>
```

method: "post" "get" The known HTML mechanisms for data return

The variable X is set to the value "1" and the URL #card is called with parameter x=1 (as #card?x=1).

The variable y is set to "2", if this link is selected.

<setvar> can be used together with <go>, <prev> and <refresh>.

## Events

Events have a Card react on certain triggers: The expiration of a timer and the entry into a card by <go> and <prev>.

```
<card ontimer="/next">
  <timer name="t" value="100"/>
  <p>
    $t seconds left until transfer to next card
  </p>
</card>
```

ontimer: URL to be called on timer expiration

value: starting value of timer in seconds

Exactly the same effect has the following example:

```
<card>
  <onevent type="ontimer">
    <go href="/next"/>
  </onevent>
  <timer name="t" value="100"/>
  <p>
    $t seconds left until transfer to next card
  </p>
</card>
```

type: "ontimer" "onenterforward" "onenterbackward", the event which should trigger the given action: Timer, <go> or <prev> entry into the CARD.

## CONCLUSION

Wap's designed is flexible for client's minimal resources such as computing power and memory storage. WAP is programmed in wireless markup language WML (application of XML) and WMLScript (WAP's version JavaScript) which is embedded in client's mobile. WML provides a simple event mechanism that allows different content to be displayed. User actions, such as pressing a key, can be tied to scripts that cause changes in content. The WML browser also has this timer function that can load a different page or trigger the change of variables when the time is up. This provide great flexibility than the static content that HTML can deliver.