What Is WAP?

WAP stands for Wireless Access Protocol, a general term used to describe the multi-layered protocol and related technologies that bring Internet content to mobile devices such as cell phones.

Such devices are referred to as thin clients because they have one or more constraints in the form of display, input, memory, CPU, or other hardware or usability limitations. The platform constraints and the slower (and more expensive) bandwidth of cellular and related networks make standard Internet protocols difficult to utilize. Using the growing set of WAP tools and protocols, however, the mobile Internet is quite a capable tool.

A Brief History of WAP

As previously stated, WAP refers to a wide range of technologies and protocols, all related to mobile Internet functionality. This functionality has roots dating back to the mid-1990s. At that time, several vendors were working on the mobile Internet problem as mobile device sales skyrocketed, and several competing technologies emerged:

Nokia’s Narrow Band Sockets (NBS) and Tagged Text Markup Language (TTML)

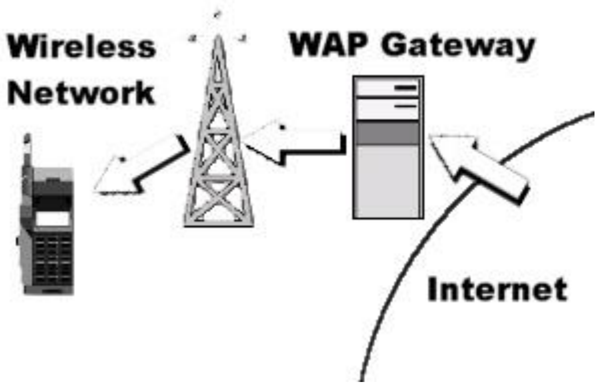
Ericsson’s Intelligent Terminal Transfer Protocol (ITTP)

Unwired Planet’s Handheld Device Markup Language (HDML)

Each technology had its own purpose, but some overlapped with others in various areas. This diversity threatened to fragment the wireless industry along provider lines. In mid-1997, the WAP Forum was founded to aid in communication among the developers and to spur a common set of protocols and technologies. In the same year, the industry took another step forward with the formation of the Open Mobile Alliance (OMA), which combined several distinct development and standards bodies into one.

How Does WAP Work?

These articles will focus on the delivery of WML content to mobile devices over a cellular or related technology network. However, the delivery of many protocols and technologies takes the same route-namely, through a proxy server that bridges the gap between the wired Internet and the wireless service provider's network.



This proxy server manages the communication between the wireless client and the Internet server(s), acting as a gateway to the wired Internet. It caches content and, in some cases, even translates raw HTML into WAP-compatible protocols such as WML.

Many mobile devices have a built-in wireless browser. Although several different browsers are in use today among the various wireless providers, most browsers support WML, either natively or translated into HDML. A popular precursor to WML, the Handheld Device Markup Language (HDML), is still supported on several mobile platforms. However, due to the limitations of HDML (supporting only a handful of navigation tags and virtually no formatting tags), WML is becoming the most widely used mobile markup language. That said, if you plan to support a particular platform, it's best to test your code extensively on that particular device.

Note: When coding for the general public, be careful to stick to the standards and avoid using proprietary extensions to the various languages, no matter how tempting the feature set of the extensions. If you decide to provide the extensions to those who can use them, you should take the necessary server steps to identify the connecting browser and deliver code customized for that browser.

What Is WML?

WML (Wireless Markup Language) is the dominant language in use with wireless devices today. Essentially, WML is a subset of HTML, but has its roots in XML. Those developers with a solid base in XML should have a relatively easy time coding WML.

The current WML standard is 1.3, although many mobile devices in use today support only the WML 1.1 standard. Therefore it’s prudent to stay away from 1.3-specific features, unless you know that your target market’s devices are 1.3-ready.

There are several key differences between WML and standard HTML, including the following:

-
  - WML is highly structured and very particular about syntax. Several current HTML browsers allow for “messy” code such as missing tags and other formatting snafus. Such mistakes are not allowed in WML; the mobile browser will complain and generally won’t display the page.

-
  - WML is case sensitive. The tags <b> and <B> are treated as different tags, although they accomplish the same purpose (bold text).

Therefore, you must be careful to match the case of your opening tags with your closing tags (for example, <b>This is bold</B> will not work as expected).

-
    - Many tags have required attributes. Developers accustomed to HTML may be used to including only attributes they need-in some WML tags, you must include a few attributes, even if they are blank or default.

- WML pages are structured in “decks” (see the next section), allowing for multiple pages to be defined in each WML file.

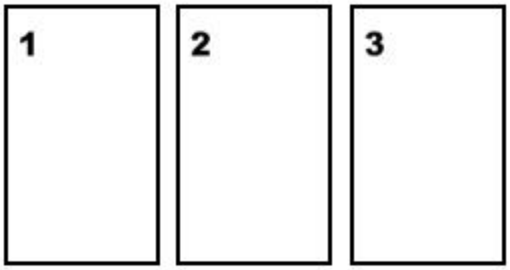
WML also has a client-side scripting language, WM Script, to help automate particular tasks, validate input, and so on.

Understanding Decks

WML pages are structured within “decks,” allowing several pages (“cards”) to be defined in each WML file. This deck analogy allows multiple pages to be delivered to the mobile client at the same time, minimizing the loading time between related pages. However, the limited memory on most devices constrains the deck size, usually to less than 1024 bytes. Therefore, careful consideration and planning should go into any WAP application; don’t start coding without investing time in planning.

Note: Remember your audience. Mobile users generally scroll through cards rapidly and will be reading on a display that’s a mere handful of characters wide (usually less than 20 characters) and usually less than 10 lines high. Keep your content to a minimum, provide an intuitive navigation structure, and optimize your decks to maximize links within the deck and minimize links outside of the deck.

Visualizing a physical “deck of cards” structure can help in understanding the principles of WML. For example, suppose we have three simple cards (pages) as shown below:



A developer accustomed to HTML might be tempted to implement the “back” feature by providing a link to the deck, specifying the previous card. However, this would cause the mobile device to re-request the whole deck before redisplaying the card-a card it already had in memory.

Instead, you should use the tag, which tells the browser to remove the current page and display the previous page in the history list (like using the Back button on a PC browser). Of course, the content of the previous page might need to be refreshed each time it’s accessed; in that case, valid techniques could include recalling the whole deck or specifying that the page not be cached.

Setting Up Your Server for WML

To configure your Web server to deliver WML, you must define the related MIME types for WML content. Web servers and client browsers use MIME (Multipurpose Internet Mail Extensions) to communicate the type of data that is being sent. Before sending data, the server sends a MIME identifier to the client browser, identifying the format of the following data. The client browser can then properly decode and apply the data. Most WML applications require three MIME types, as listed in the following table.

.wmltext/vnd.wap.wmlWML source file
.wmlstext/vnd.wap.wmlscriptWML script file
.wbmpimage/vnd.wap.wbmpWireless bitmap file (image)