

WAP Development Toolkits

Dec 2000

Abstract

This document provides a brief introduction on the mobile application market size and opportunities for provisioning mobile Internet services. The paper discusses briefly the Wireless Application Protocol (WAP), Wireless Markup Language (WML) and WMLScript. The document then discusses the WAP development toolkits from Nokia, and phone.com. The document briefly discusses other development platforms such as the Palm Emulator, Ericsson Toolkit and the IBM's Wapsody.

Introduction

The unparalleled growth of the Internet has been accompanied by a strong growth in the use of cellular phone. The cellular phone and the Internet market are rapidly converging. It is estimated that there are currently over 285 million cell phone users in the world [1]. The predictions indicate that there will be 525 million WAP enabled handsets by 2003. The user base accessing mobile Internet services is also expected to grow from 2 million users in 1999 to over 12 million users by 2001 [1].

There is a demand for providing consumers with wireless applications that can be easily accessed on their cell phones or their PDA's. Consumers are demanding services such as

- ?? Real time delivery of content such as news, stock quotes, sports scores.
- ?? Personal productivity tools such as email, personal organizers.
- ?? E-Commerce applications allowing users to purchase products and services, access to online auctions etc.
- ?? Access to Business Applications.

The demand for wireless services meant that a large number of corporations and startup began developing new protocols and applications. One of the largest groups to emerge from this was the WAP Forum started by phone.com. Three large cell phone manufacturers, Ericsson, Motorola and Nokia joined this group. The WAP Forum began defining standards for providing Internet access to wireless devices. The WAP Forum now has over 200 members. The WAP Forum publishes these standards as WAP specifications.

Wireless Application Protocol Specification

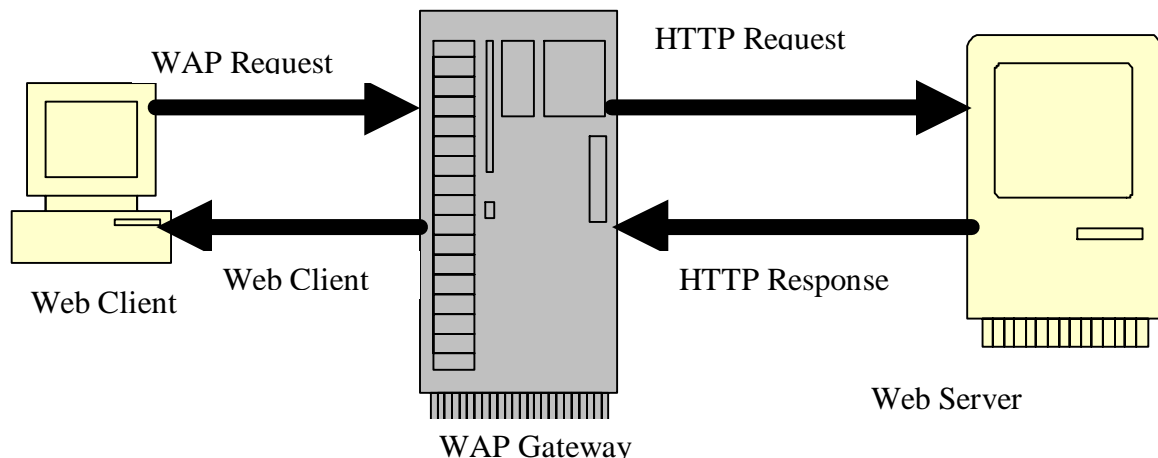
The current WAP specification from the WAP Forum is version 1.2. The WAP specification has the following key components.

- ?? A programming model and development environment that was based on the existing Web programming model. The decision was important because it allowed corporations to leverage their existing skills.
- ?? A lightweight protocol stack. This was important because existing wireless devices and protocols have low bandwidth. Compression and security are key issues in the design of the protocol stack.
- ?? An XML-based markup language designed to create WAP applications. The browser based application model has become very dominant today. The WAP Forum's first design goal was to have a web-based programming model. The XML-based language was called the Wireless Markup Language (WML) and is similar to HTML.
- ?? A scripting language for performing some additional computations. The scripting language that the WAP Forum developed was WMLScript and is very similar to other client side scripting languages such as VBScript and Javascript.
- ?? A micro-browser specification that defines how WML and WMLScript be displayed in the handheld device.

The next sections discuss in brief the key ideas of the WAP specification.

Web Programming Model

The WAP programming model is based on the WWW programming model or a transactional processing model. Resources are located through a Uniform Resource Locator (URL). With a WAP programming model, an intermediary is required to transform WAP requests into HTTP requests and HTTP results back into WAP results. This intermediary is the WAP Gateway.



Nokia and phone.com are companies that provide WAP gateway software. The WAP gateway also provides additional services such as email services etc.

Wireless Markup Language (WML)

The Wireless Markup Language (WML) is used to develop applications for creating applications that will be executed on mobile devices such as cell phones and personal digital assistants (PDA's). WML is an Extensible Markup Language (XML) Data Type Definition (DTD). WML is similar to HTML and a large number of the tags sets from HTML are available in WML. WML has the following general capabilities

- ?? Support for formatting text and images.
- ?? Support for navigating within pages and to external pages.
- ?? Support for accepting user inputs.
- ?? Support for multiple languages.
- ?? State and content management features.

Decks and Cards:

A key idea in WML is the use of cards and decks. WML files are organized as decks and is the content that gets loaded into the WAP device. WML decks are organized as cards and only one card is displayed on the screen at a time. WML provides facilities for movement back and forth between cards, and for loading completely new WML decks. The figure below show WML code and the resulting output

```
<wml>
<card>
  <p align="center">
    Pittsburgh
  </p>
  <p mode="nowrap">
    <table columns="2">
      <tr>
        <td> <i>High</i> </td>
        <td> <i>Low</i> </td>
      </tr>
      <tr>
        <td>83</td>
        <td>45</td>
      </tr>
    </table>
    Today: Sunny. <br/>
    Tonight: Cold with chance of light rain.
  </p>
</card>
</wml>
```



Table 1:WML Code and display output

WMLScript

HTML clients benefited by having a client-side scripting module built in. Netscape developed Javascript and Microsoft developed VBScript. The WAP Forum also decided that having a scripting language in the specification would be helpful. They added WMLScript to the model. WMLScript is a scaled down version of existing scripting languages. Features that were necessary for wireless application development were retained.

WAP Micro Browser:

The WAP microbrowser is similar to any regular HTML browser and has features to submit requests and display responses. It has features to display forms, tables and images. The microbrowser has built in WML and WMLScript interpreters. The WAP browser is also able to communicate with the WAP protocol stack. The browser can have additional functionality such as caching, history stack etc. The browser has to be light-weight since it has to work with limited memory, small screen size, limited input and output capabilities.

Wireless Application Protocol Development Toolkits and Platforms

The demand for wireless application has resulted in a number of development toolkits. These toolkits allow developers to develop applications, test them on a local machine and move them to a WAP enabled server. There are a number of toolkits on the market place from leading manufacturers like Nokia and Ericsson. However the two most popular ones appear to be the Nokia Toolkit and the phone.com SDK. The rest of the document will discuss these toolkits.

Nokia Toolkit:

The Nokia WAP Toolkit is a Java based software toolkit that enables developers to develop wireless applications. The tool kit provides a number of components such as

- ?? Nokia Handset Simulators with built in WML browser
- ?? WML and WMLScript Editors
- ?? Sample Applications

The toolkit can be downloaded from the Nokia website after a simple registration process. The toolkit is available as an installable set of files. The installation process is extremely easy and the selected components are installed.

The toolkit offers the following components

- ?? An editor that can be used to compose WML/WMLScript and WBMP files. The editor is a very simple editor that can be used to write and compile WML code. It provides some elegant features such as color-coding of the WML tags. WBMP's can be created and edited in the toolkit.

- ?? Nokia cell phone emulators. The toolkit offers the developer two cell phone simulators.
- ?? Documentation – The documentation that comes long with the toolkit is excellent and includes
 - o Developer Guide
 - o Installation Guide

The figure below shows the Nokia Toolkit.

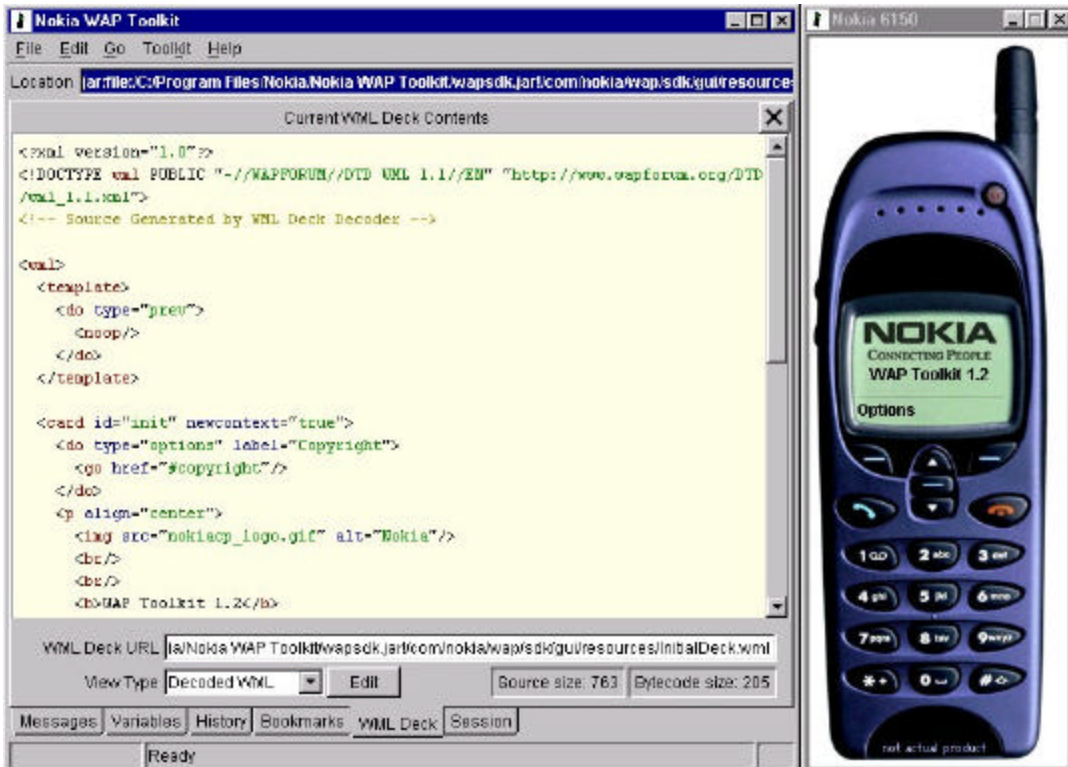


Figure 1: Nokia Toolkit

Application developers can compose and test their applications on emulators and expect them to work the same way on the actual devices. It provides an easy environment for developers to write, test and debug applications on a PC-based simulator.

The toolkit implements Wireless Application Protocol (WAP) and provides tools to write and compile WML and WMLScript. The simulator gives a real-time WYSIWYG depiction of the content on a WAP enabled handset. The developer also gets detailed information about the application from the debugging tools.

Phone.com Software Development Kit:

The Phone.com UP.SDK [5] product differs a bit from the Nokia Toolkit. It is a native application rather than a Java application. It does not have an IDE. Developers use other tools to write the code. The SDK focuses much more heavily on providing a set of reusable code libraries for use with languages such as WML, Perl, C, C++, and Visual Basic. The SDK is aimed at targeting experienced developers rather than beginners. The UP.SDK also comes with the a WAP browser known as the UP.Simulator, as shown in the figure below.



Figure 2: UP.SDK Microbrowser

The SDK comes with excellent documentation and a large number of sample programs. There are a number of API's and tools available to developers such as

- ?? Perl and C library functions for generating WML and handling HTTP requests
- ?? C++ objects for Solaris and COM objects for Windows.
- ?? Asynchronous notifications can also be pushed to handheld clients via the Phone.com Notification API. This API allows the control of alerts, document cache, and decks on the client.
- ?? Tools for requesting and installing SSL certificates for security purposes.

Comparison Of Nokia WAP Toolkit and Phone.com SDK:

	Nokia Toolkit	Phone.com SDK
Platforms	Windows NT, Requires Java 2	Windows 95/98/NT and Solaris
Size of Download	7 MB	6 MB
Memory Requirements	64 MB	32 MB
WAP Browser	Good, but has problems rendering certain WML	Good browser. Consistent in its display and rarely gives problems in rendering WML.
Editor	Editor Support WML, WMLScript and WBMP.	No WML Editor
Documentation	Excellent documentation on WML, WMLScript and the Toolkit.	Excellent documentation especially for advanced developers.
Security Support	No security tools	Provides the CertMaker tool to create and install certificates.
Image Editor	Provided with the WML editor	No image editor
Operating Modes	HTTP and through WAP Gateway	HTTP and through WAP Gateway
Debugging	Debugging Window with error messages	Debugging Window with error messages
Compiler	WML code can be compiled to generate binary compiled files.	Compiler not provider

Ericsson WAP IDE:

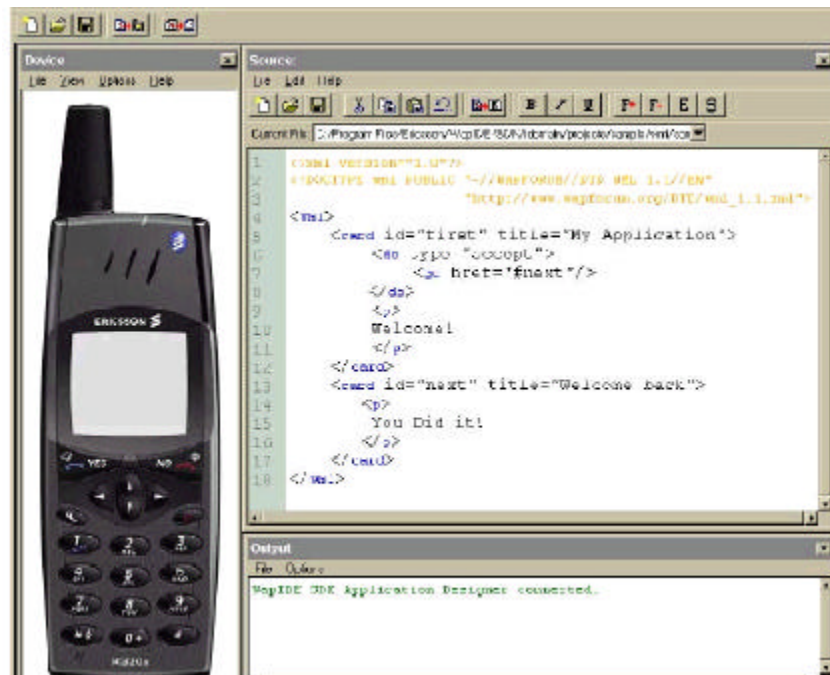
The Ericsson WapIDE product consists of a suite of tools that support the design and testing of WAP applications. This toolkit is similar to the Nokia toolkit. WapIDE Software Development Kit is currently only available the Windows platform. The toolkit consists of three tools

??The Application Designer – is the IDE that allows developers to create, compile and test WML and WMLScript applications.

??The Browser – allows the developer to view and test WML decks and cards using a simulated WAP device.

??The Server Toolset – these include tools such as compilers and libraries.

The Ericsson WAP IDE is shown below and resembles the Nokia Toolkit.





IBM Wapsody:

IBM's Wapsody [8] is different from the toolkits that we have seen in the earlier sections. Wapsody is a Java package that implements most layers of the specification. Wapsody simulates most aspects of WAP. Wapsody is designed for use in a WAP application-building environment. Wapsody classes can be used by developers to build any Java application ranging from front-end Java clients to server side programs. Wapsody classes could be used to develop Java applications that run on hand held devices that have a Java VM. These devices need not have a WAP browser built in.

The biggest problem with employing Wapsody on handheld devices is the heavy resource requirement. It appears at the moment that Wapsody is not ready to be deployed on handheld devices, but can be used on Windows 95/98 platform. This means that Wapsody could be used on wireless PC's that do not use TCP/IP, but some wireless protocol as the underlying transport mechanism. The documentation for Wapsody is also unavailable.

Conclusion

In my opinion, both the Nokia WAP toolkit and the phone.com SDK are good tools for developing and testing WAP applications. The Nokia WAP Toolkit product is a better starting point since it comes with excellent documentation as well as an easy-to-use graphical development environment. The resource requirements are quite heavy. The Phone.com SDK is a much leaner product and offers a great deal more capabilities but does not have any development tools. It has excellent documentation too and has more API's for advanced developers. Developers will benefit by installing both the toolkits when developing their applications. There other toolkits provide a wide range of facilities for developers, but the two toolkits discussed in this document are probably the best toolkits.

Reference:

- [1] phone.com – *Enabling the Wireless Internet* [<http://www.phone.com>]
- [2] Steve Mann - *Programming Applications with the Wireless Application Protocol*, John Wiley & Sons Inc [ISBN-0-471-32754-9]
- [3] Keith Rollin – *Palm OS Emulator 2.1/3.0* [<http://www.palm.com>]
- [4] Au-System Website – [<http://www.ausystems.com>]
- [5] Phone.com Website – [<http://www.phone.com>]
- [6] Nokia WAP Website – [<http://www.nokia.com/wap>]
- [7] A First Look At WAP Toolkits From Nokia, Ericsson, and Phone.com
- [8] IBM Wapsody Website - <http://alphaworks.ibm.com/aw.nsf/techmain/wapsody>