



Ordering Information: [Internet & World Wide Web How to Program, 2/e](#), [The Complete Internet & World Wide Web Training Course, 2/e](#), [Wireless Internet & Mobile Business How to Program 1/e](#) & [The Complete Wireless Internet & Mobile Business Training Course, 1/e](#).

- View the complete **Table of Contents**: [Internet & World Wide Web How to Program, 2/e](#) & [Wireless Internet & Mobile Business How to Program, 1/e](#)
- Read the **Preface**: [Internet & World Wide Web How to Program, 2/e](#) & [Wireless Internet & Mobile Business How to Program, 1/e](#)
- Download the **Code Examples**: [Internet & World Wide Web How to Program, 2/e](#) & [Wireless Internet & Mobile Business How to Program, 1/e](#)

To view all the Deitel products and services available, visit the Deitel Kiosk on InformIT at www.informIT.com/deitel.

To follow the Deitel publishing program, sign-up now for the *DEITEL™ BUZZ ONLINE* e-mail newsletter at www.deitel.com/newsletter/subscribeinformIT.html. To learn more about our [Internet and or Wireless Internet programming courses](#) or any other Deitel instructor-led corporate training courses that can be delivered at your location, visit www.deitel.com/training or contact Christi Kelsey, Director of Corporate Training Programs, at (978) 461-5880 or e-mail: christi.kelsey@deitel.net.

Note from the Authors: This article is an excerpt from Chapter 23, Sections 23.9, 23.10 and 23.11 of *Internet and World Wide Web How to Program, 2/e*. This article discusses the Wireless Markup language (WML), the markup language used to create presentational layer of WAP applications delivered to wireless handheld devices. This first article in a two-part series presents an introduction to WAP and WML. In Part 2, we introduce the WMLScript scripting language. Readers should be familiar with basic markup concepts (e.g., HTML, XML) and Internet Information Services (IIS) or Apache Web. The code examples included in this article show readers examples using the Deitel™ signature *LIVE-CODE™ Approach*, which presents all concepts in the context of complete working programs followed by the screen shots of the actual inputs and outputs.

23.9 WAP and WML

One of the most important aspects of wireless communications is standardization. In 1997, the Wireless Application Protocol (WAP) was developed by dominant cell-phone manufacturers Nokia, Ericsson, Motorola and others to facilitate the introduction and standardization of wireless Internet access.¹⁹ WAP is a set of communications protocols that are designed to enable wireless devices to access the Internet. WAP applications can be used on Palm OS, Windows CE, Mac OS and Java 2 Micro Edition devices.²⁰

Although WAP communications involve many components, we focus on three—a *WAP-enabled mobile device*, a *WAP gateway* and a *Web server*. When a user of a WAP-enabled device requests information from the Internet, the device sends the request to a WAP gateway. WAP gateways serve as links between mobile devices and the Internet. WAP gateways are designed to convert Web content from WML to HTTP, which is the standard protocol used to transfer and view information in Web transactions. The WAP gateway communicates with the Web server (i.e., the server that has a connection to the Internet). The Web server processes the mobile-device request by searching through existing databases and information resources, such as Web pages. The Web server then transmits the requested information back to the WAP gateway, using HTTP. The gateway translates the information into WML and sends it to the mobile device for use.²¹

The Wireless Markup Language (WML), which is based on XML, is the markup language used to create Web content delivered to wireless handheld devices. *WML tags* are the markup commands that specify how a Web page should be formatted for viewing on a wireless device. *Microbrowsers*, browsers designed with limited bandwidth and memory requirements, can access the Web via the wireless Internet. WAP supports WML to deliver the content.

A WML document is called a *deck*; each contains one or more pages, called *cards*. Cards are renderable units of WML documents useful for WAP clients (a WAP client being any WAP-enabled device) that generally use the devices with limited screen sizes. Each card can contain both text content and navigational controls to facilitate user interaction. Though only one card can be viewed at a time, navigation between cards is rapid, because the entire deck is stored by the microbrowser.²²

Although WAP and WML have many advocates, they also have many opponents. Those who favor WAP technology see it as a short-term solution for the delivery of wireless Internet access. WAP opponents cite various disadvantages that are associated with the protocol, including possible security breaches, limited bandwidth and unreliability.

The limited bandwidth capabilities of WAP-enabled devices cause a host of problems. Not only are WAP-enabled devices unable to handle the transmission of multimedia, but they can also become overloaded during peak hours, the busiest hours of the day for conducting wireless communications.²³ This limitation causes business-to-business (B2B) and business-to-consumer (B2C) application developers to anticipate the release of faster 3G technologies and a new WAP specification that supports increased functionality.

When learning about mobile communications, it is vital to understand the process by which mobile devices connect to and interact with the Internet, because this process is organized differently for each protocol and programming language. However, no protocol currently existing allows a wireless device to communicate directly with the Internet. Each system (i.e., WAP and WML, i-mode and Java and J2ME) employs its own method of sending and receiving information to and from the Internet.

23.10 Phone Simulator and Setup Instructions

Several free browsers are available for the development and testing of WAP applications. Openwave and Nokia® are the two most popular WAP browsers. The Openwave Software Development Kit is available at

`developer.openwave.com/download/license_41.html`

and the Nokia Wireless Toolkit is available at

`http://www.forum.nokia.com/not_logged?path=/wapforum/main/1,35452,1_1_50,00.html`

The Openwave Simulator is part of Openwave's *Software Development Kit (SDK)*, a group of tools for wireless developers. The Openwave Simulator replicates the behavior of the Openwave browser that is used on actual wireless devices. Openwave's *UP.SDK Getting Started Guide*, which facilitates the installation and use of the SDK, is available at `developer.openwave.com/html/doc/41/getstart`.

In this chapter, we use the Openwave UP.Simulator. To install the Openwave UP.SDK on a machine running Windows 2000, perform the following steps: [*Note: More detailed installation instructions are available on the Deitel & Associates, Inc., Web site, www.deitel.com.*]

1. Go to the Openwave Web site, and click the **Download** button.
2. Check the **Save this program to disk** option, and click **OK**.
3. Select a name and location for the file, and click **OK**.
4. Once the document is complete, click **Open** to begin the installation process.
5. When the **Welcome** screen of the installation program appears, click **Next**.
6. Read the license agreement, and click **Yes**.
7. Read the **SCREENSHOTS AND IMAGE USE AGREEMENT**, and click **Yes**.
8. Read the text of the **Safe Country Verification** dialog, and check the **Yes** box before clicking **Next**.
9. Choose a destination folder in which to install the Openwave browser by clicking the **Browse** button, or select the default folder (recommended). Click **Next** to continue.
10. Specify the name of the folder to appear in the **Start** menu, and click **Next**.
11. Setup is now complete. To view the **README** file and launch the Openwave browser (UP.Simulator), check each respective option. Click **Finish**.

To simulate WAP applications by using the Openwave and Nokia browsers, WAP documents must be requested from a Web server, such as *Apache* or *Internet Information Services (IIS)*. Visit the **Downloads/Resources** page on our Web site (www.deitel.com) to access Web server installation and configuration instructions. The following instructions cover the configuration of the IIS Web server. Apache configuration instructions can be found at our Web site.

In IIS, copy the example files to **C:\Inetpub\wwwroot**. Alternatively, these files can be placed in a directory referenced by a virtual directory. A virtual directory is an alias for an existing directory that resides on the local machine (e.g., **C:**) or on the network.

To configure IIS to respond to requests for WAP documents, perform the following:

1. Right click the Windows task bar, click **Properties** and select the **Advanced** tab. Under **Start Menu Settings**, select **Display Administrative Tools**, and click **OK**.
2. Click the **Start** button on the Windows task bar and select **Programs**, then **Administrative Tools**, then **Internet Services Manager**.
3. In the **Internet Information Services** dialog, double click the computer name, or click the plus sign next to the computer icon to expand the list.
4. Right click **Default Web Site**, and select **Properties**.
5. Click the **HTTP Headers** tab, and select the **File Types...** button, which is located within the **(MIME): Map Frame Settings** box.
6. Click the **New Type** button.
7. In the **Associate Extension** text box, type **wml**. In the **Content type {MIME}** text box, type **text/vnd.wap.wml**. This allows the server to map WML documents to the **.wml** file extension.
8. Click **OK**.
9. Click the **New Type** button.
10. In the **Associate Extension** text box, type **wmls**. In the **Content type (MIME):** text box, type **text/vnd.wap.wmlscript**. This allows the server to map WMLScript files to the **.wmls** file extension. WMLScript facilitates a disciplined approach to the designing of programs that enhance the functionality of WML documents. The relationship between WML and WMLScript is similar to that between XHTML and JavaScript.
11. Click **OK**.

To request a document, it is necessary to launch the Openwave Simulator. This is achieved by typing **localhost/fileName.wml**, in the Openwave Simulator's **Go** field and pressing **Enter**. For example, to request our first WML document (Fig. 23.2), type **localhost/fig23_2.wml**. If the document is located in a **wwwroot** subdirectory or in a virtual directory, the folder name or virtual directory name must precede the document's file name (e.g., **localhost/folderName/fileName.wml**).

23.11 Creating WML Documents

In this section, we begin to create WAP applications by marking up information using WML. Figure 23.2 presents a WML document that displays a welcome message. The screen shot of the Openwave Simulator displays the WML document. The **Phone Infor-**

mation window below the simulator displays the status of the simulator. If an error occurs during the rendering of the document, the error is listed in this window.

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
3   "http://www.wapforum.org/DTD/wml12.dtd">
4
5 <!-- Fig. 23.2: fig23_2.wml -->
6 <!-- Simple WML Page -->
7
8 <wml>
9   <card id = "index" title = "WML Title">
10    <p>
11      Welcome to wireless programming!
12    </p>
13  </card>
14 </wml>
```

Fig. 23.2 Simple WML document (part 1 of 3). (Image of UP.SDK courtesy of Openwave Systems Inc. Openwave, the Openwave logo, and UP.SDK are trademarks of Openwave Systems Inc. All rights reserved.)



Fig. 23.2 Simple WML document (part 2 of 3). (Image of UP.SDK courtesy of Openwave Systems Inc. Openwave, the Openwave logo, and UP.SDK are trademarks of Openwave Systems Inc. All rights reserved.)

```

#Phone Information
<http://updev.phone.com/dev/wml/devhone4.wml>
)
HTTP GET Request: http://updev.phone.com/dev/wml/devhone4.wml
Checking for URL 'updev.phone.com/dev/wml/devhone4.wml' in Whitelist
not found

----- DATA SIZE -----
Uncompiled data from HTTP is 1109 bytes.
...found Content-Type: text/vnd.wap.wml.
Compiled WAP binary is 537 bytes.

cache miss: <http://localhost/fig23_2.wml>
net request: <http://localhost/fig23_2.wml>

HTTP GET Request: http://localhost/fig23_2.wml
Checking for URL 'localhost/fig23_2.wml' in Whitelist
not found

----- DATA SIZE -----
Uncompiled data from HTTP is 1492 bytes.
...found Content-Type: text/vnd.wap.wml.
Compiled WAP binary is 168 bytes.

```

Fig. 23.2 Simple WML document (part 3 of 3). (Image of UP.SDK courtesy of Openwave Systems Inc. Openwave, the Openwave logo, and UP.SDK are trademarks of Openwave Systems Inc. All rights reserved.)

WML is an *XML vocabulary* (i.e., a markup language that is created using XML). Line 1 is the optional XML declaration that specifies the **version** of XML to which this document's syntax adheres. Lines 2–3 specify the Document Type Definition (DTD) to which the document conforms. The root element (i.e., the document's first element) for every WML document is **wml**.

Line 9 contains the opening **card** element. The **card** element has two important attributes: **id**, which uniquely identifies the card, and **title**, which displays a title at the top of most microbrowser windows. The Openwave browser does not display **card titles**.



Common Programming Error 23.1

WML element and attribute names are case sensitive and must be written in lowercase. Writing a WML tag name in uppercase is an error.



Common Programming Error 23.2

Every WML document requires a minimum of one **card** element, which contains information (e.g., text, images or links). Failure to include this element is an error.



Good Programming Practice 23.1

A wireless device's display is often limited, so keep **card titles** concise.



Portability Tip 23.1

Some microbrowsers do not display the value of the **title** attribute.

All text in a WML document is placed between **<p>** tags that are nested within **<card>** tags. Although the text occupies only one line in the document, the screen capture

in Fig. 23.2 shows the text displayed on two lines. When a line of text exceeds the width of the display, the Openwave browser wraps the text onto the next line.



Portability Tip 23.2

Not all browsers wrap text onto the next line of the display window. In some cases, long lines of text run off the right side of the screen. Many devices do not support horizontal scrolling; therefore, the text cannot be read. Always test WAP applications on devices on which these applications are likely to run.

Figure 23.2 also includes a screen capture of the **Phone Information** window. This window alerts the developer to any errors that occur during the testing of WML and WML-Script documents. For example, if we had left out the `</card>` end tag (line 13 of Fig. 23.2), the Openwave Simulator would display the error message, “**Compile Error. See Info Window for Details.**” in the display window. Figure 23.3 shows a screen capture of the **Phone Information** window detailing the specifics of the error. Wireless devices do not contain this window.

```

Phone Information
----- DATA SIZE -----
Uncompiled data from HTTP is 1479 bytes.
...found Content-Type: text/vnd.wap.wml.

***** WML Errors *****
WML translation failed.
(15) : error: Close tag 'wml' does not match start tag 'card'
(32) : error: Expected the end of root element instead of end of file

***** End Errors *****

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

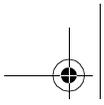
<!-- Fig. 23.2: fig23_2.wml -->
<!-- Simple WML Page -->

<wml>
  <card id = "index" title = "WML Title">
    <!-- Every card must include a paragraph element -->
    <p>
      Welcome to wireless programming!
    </p>
  </wml>

```

Fig. 23.3 Phone Information window showing an error in the deck.

The section labeled **WML Errors** lists each error, along with an accompanying line number. Below the list of errors, the window lists the WML code. Note that in this example, the `</card>` end tag is missing.



WORKS CITED

The notation <www.domain-name.com> indicates that the citation is for information found at that Web site.

19. T. Hughes, "The Web Unwired," *Global Technology Business* December 1999: 33.
20. Fixing WAP's Security Flaw," *m-business* January 2001: 92.
21. <www.wapuseek.com/wapfaqs.cfm#4>.
22. <www.wapuseek.com>.
23. S. Phan, "Who Needs a PC?" *Business 2.0* 14 November 2000: 54.