**CHAPTER**

# 1

# Why Bother?

**TOPICS IN THIS CHAPTER**

**E**xtensible Hypertext Markup Language, better known as XHTML, is among the latest and greatest technologies to hit the Web. Developed by the World Wide Web Consortium (W3C) as the official successor to the tried-and-true but aging Hypertext Markup Language (HTML), XHTML marries the strength of HTML's formatting and presentation capabilities with the red-hot Extensible Markup Language (XML) that is today revolutionizing Internet data representation and information exchange. In short, XHTML is a reformulation of HTML in terms of an XML language that promises to bridge the wide gap between yesterday's HTML-based eye candy Web pages and tomorrow's industrial-strength Web applications. XHTML is the official successor to HTML, which was recently laid to rest when the W3C released XHTML 1.0 in January 2000. Developed through the W3C's HTML Working Group (a formal process group that drives the design and development of HTML, as you'll learn in Chapter 2, "XHTML = HTML + XML"), XHTML 1.0 was designed to bring the rigor of XML to Web pages while accommodating an everchanging end user landscape.

According to the W3C HTML Working Group, "*XHTML 1.0 is the keystone in W3C's work to create standards that provide richer Web pages on an everincreasing range of browser platforms including cell phones, televisions, cars, wallet-sized wireless communicators, kiosks, and desktops. XHTML is modular, making it easy to combine with markup tags for things like vector graphics, multimedia, math, electronic commerce, and more. Content providers will find it easier to produce content for a wide range of platforms, with better assurances as to how the content is rendered. The modular design reflects the realization that a one-size-fits-all approach will no longer work in a world where browsers vary enormously in their capabilities.*"

Tim Berners-Lee, inventor of the Web and W3C founder and Director, adds, "*XHTML 1.0 connects the present Web to the future Web. It provides the bridge to page and site authors for entering the structured data XML world, while still being able to maintain operability with user agents that support HTML 4.*"

Although XHTML is new, it holds great promise for the future of the Web. Internet experts from around the world agree that XHTML is the bridge that brings us to the next generation Web without burning our bridges to the past (see "XHTML Testimonials," later in this chapter, for details).

But really, why bother? Wasn't XML itself supposed to be the killer technology that enables our next generation Web sites? XML, after all, is a panacea for the Web's ills, isn't it? When it first arrived on the scene a few years ago, XML promised to usher in the age of the *semantic Web*, an age where well-formed, "validated" content enables automated data exchange and next generation Web sites. XML, according to many industry pundits, promises to obviate the need for HTML altogether in the near future. So why bother with a middleman like XHTML when you can go straight to XML?

Why not construct your current and future Web applications around XML lock, stock, and barrel, leaving HTML (and its newly anointed successor, XHTML) in the dust? If XHTML bridges the gap between HTML and XML, why not leap over that gap altogether and jump directly into XML today? In doing so, wouldn't you save yourself the hassle of learning yet another markup language, spare yourself the pain and suffering of developing, testing, and deploying Web content in yet another new and relatively unproven cutting-edge technology, and save yourself a few bucks on the cost of this book? The answer is no.

 XML, in and of itself, is simply a *metalanguage* that lets you define other languages, of which XHTML is just one example. XHTML is an *application of XML* designed specifically for building Web pages. As the next evolution of HTML, XHTML is the language we'll use to construct cutting-edge Web sites now and in the foreseeable future. When you use XHTML you are, in fact, using XML. Because the two are inseparable, you can't skip over XHTML and go directly to XML, no matter how hard you might try.

Furthermore, you can't ignore HTML altogether when you use XHTML because the latter is based on the former. Whereas XHTML is an application of XML, XHTML gets its *meaning* from HTML. In other words, XHTML is HTML expressed as an XML application. Or, as the W3C puts it, XHTML "*is a reformulation of HTML 4.01 in XML, bringing the rigor of XML to HTML, and can be put to immediate use with existing browsers by following a few simple guidelines.*"

Take your pick. Either way both definitions boil down to the same thing: XHTML extends the life of HTML using XML. And it's ready to be used in your Web site now. You can convert your existing HTML pages into XHTML this very moment because XHTML is backward compatible with HTML and future compatible with other XML languages. In short, XHTML is the best of both worlds.

> **XML is not an application language. It is the set of rules by which one designs an application language such as XHTML. XHTML is an application of XML that gets its meaning from HTML. XHTML is the next major evolution of HTML, and as such will dominate as a Web development markup language for the foreseeable future.**

**Note**

Because XHTML is backward compatible with HTML, properly constructed XHTML pages can be delivered to standard Web browsers today. You can tap into many of the advantages that XML offers by way of XHTML. This book shows you how.

As you'll soon see, XML is just one piece of a very large puzzle, a puzzle in which HTML and XHTML fit tongue-and-groove, as illustrated in Figure 1–1. In this book we'll piece together the sometimes mysterious and complicated Web development puzzle, with a strong emphasis on XHTML Web page development. We'll see how critical technologies such as HTML, XHTML, and XML come together—along with several other W3C technologies—to create sophisticated Web content that we could only dream about a few years ago.

In particular, you'll learn why you should convert your existing HTML pages to XHTML today, and how to go about it once you've made the commitment to do so. You'll also learn how to create static and interactive XHTML Web pages from scratch, and how to enrich them with various XML-based languages such as Synchronized Multimedia Integration Layer (SMIL), Scaleable Vector Graphics (SVG), Math Markup Language (MathML), and the Extensible 3D (X3D) language.

Finally, you'll learn how to prepare your Web site for the rapidly emerging mobile invasion by creating XHTML content that can be deployed across a variety of mobile devices, including Web-enabled cellular phones and Personal Digital Assistants (PDAs). Along the way you'll learn how to code XHTML by hand, as

well as how to construct XHTML Web pages using a variety of authoring, validation, and optimization tools.
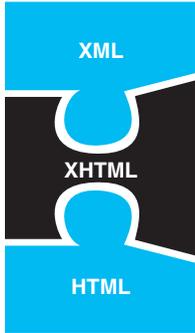
**FIGURE 1–1** HTML, XHTML, and XML are three interlocking pieces of the more extensive Web application development puzzle.

**Note**

The World Wide Web Consortium (W3C) was established in late 1994 with a charter to lead the World Wide Web to its full potential by developing common protocols that promote its evolution and ensure its interoperability. Founded by Tim Berners-Lee, the inventor of the Web, W3C is physically located at the Massachusetts Institute of Technology Laboratory for Computer Science (MIT/LCS) and digitally at http://www.w3c.org/ or http://www.w3.org/ (throughout this book we'll refer to the latter URL). A truly global organization, W3C is comprised of over 400 member organizations from around the world. Created in collaboration with CERN, the European laboratory for particle physics where Berners-Lee first conceived of the Web, W3C (with support from DARPA, the U.S. Defense Advanced Research Project Agency, and the European Commission) has already developed more than 20 technical specifications related to the Web, including HTML, XML, and XHTML. As you'll learn in the next chapter, W3C specifications are developed through collaborative member interaction that primarily takes place through e-mail, teleconferences, and face-to-face meetings hosted by the organization.

## Piecing Together the Web Development Puzzle

When the World Wide Web first burst onto our desktop computer screens more than a decade ago nobody could have anticipated how great its impact would ultimately be. What was once a novelty has blossomed into a full-blown global information infrastructure that has changed the very face of modern day life, love, and work. Although sophisticated Web applications were years away at the time (light years away, when measured in "Internet time"), even the very first Web pages created were built atop a solid foundation of three interlocking technologies that today provide the fundamental framework for our most sophisticated Web development projects.

As a Web developer, you are undoubtedly familiar with the following three foun-
dation technologies that first surfaced when Tim Berners-Lee introduced them
along with his World Wide Web "invention" to an unsuspecting Internet commu-
nity in 1991:

▼  **HTML (Hypertext Markup Language)** is the *lingua franca* of the Web. As
the glue that holds together every Web page, HTML is a mercifully simple
text-based markup language designed specifically for writing hypertext
documents that live on the World Wide Web. Based on the more sophisti-
cated Standard Generalized Markup Language (SGML) that had estab-
lished a following in the traditional publishing industry years earlier,
HTML made it easy for developers of all levels to create and publish docu-
ments that could be viewed by any Internet user armed with a Web
browser. Example 1–1, for example, contains the HTML code that corre-
sponds to the Web page shown in Figure 1–2 (Example 1–2 shows the
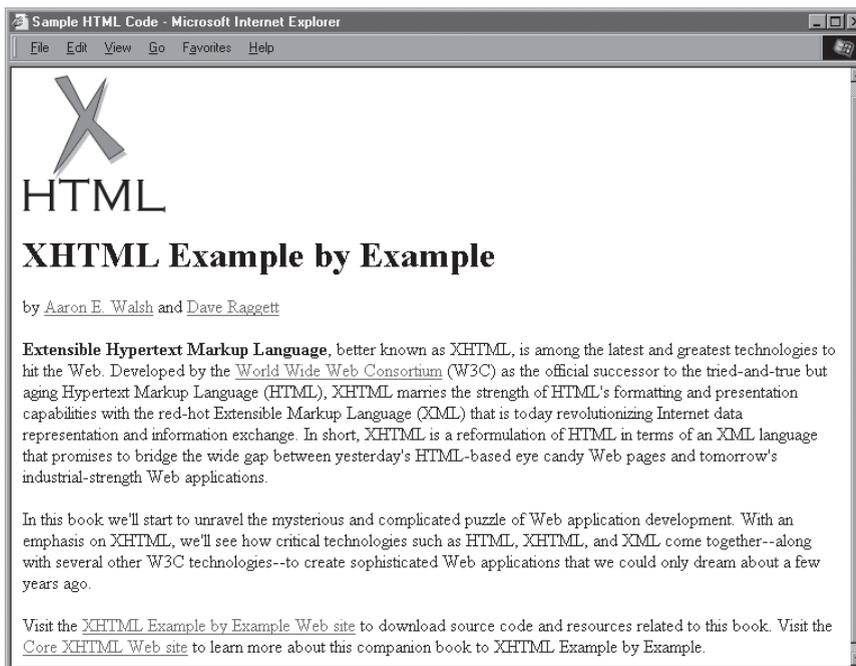
**FIGURE 1–2**  The Web page shown here contains an image, text, and
several hyperlinks, the HTML source code for which is found in Exam-
ple 1–1, while the XHTML code is shown in Example 1–2. Visit the *XHTML
Example by Example* Web site at http://www.Web3Dbooks.com/ to view
this Web page in your own browser.

equivalent XHTML code for this page, which should look familiar to you, as XHTML is based on HTML).

▼ **URL (Uniform Resource Locator)** packs the muscle behind HTML's punch; without URLs, Web pages couldn't be hyperlinked together, nor could disparate "resources" (sounds, images, movies, other Web pages, and so forth) be composed together into a single page. As an easy-to-use variation of the more sophisticated Uniform Resource Identifier (URI), URLs gave Web developers a simple yet effective mechanism for identifying and referencing content. The image and hyperlinks seen in Figure 1–2, for example, are coded into the corresponding HTML source code (Example 1–1) as URLs (shown in *blue* to make them easy for you to identify).

**Example 1–1   Sample HTML Code**

```
<HTML>

<HEAD>
<TITLE>Sample HTML Code</TITLE>
</HEAD>

<BODY BGCOLOR="white">

<P><IMG SRC="images/logo.png" ALT="logo" WIDTH="128" HEIGHT="128"></P>

<H1>XHTML Example by Example</H1>

<P>by <A HREF="http://www.mantiscorp.com/people/aew/">Aaron E.
Walsh</A> and <A HREF="http://www.w3.org/People/Raggett/">Dave
Raggett</A></P>

<P><B>Extensible Hypertext Markup Language</B>, better
known as XHTML, is among the latest and greatest technologies to
hit the Web. Developed by the <A HREF="http://www.w3.org/">World
Wide Web Consortium</A> (W3C) as the official successor to the
tried-and-true but aging Hypertext Markup Language (HTML), XHTML
marries the strength of HTML's formatting and presentation
capabilities with the red-hot Extensible Markup Language (XML)
that is today revolutionizing Internet data representation and
information exchange. In short, XHTML is a reformulation of HTML
in terms of an XML language that promises to bridge the wide gap
between yesterday's HTML-based eye candy Web pages and tomorrow's
industrial-strength Web applications.</P>

<P>In this book we'll start to unravel the mysterious and complicated
puzzle of Web application development. With an emphasis on XHTML,
```

we'll see how critical technologies such as HTML, XHTML, and XML
come together--along with several other W3C technologies--to
create sophisticated Web applications that we could only dream
about a few years ago.</P>

<P>Visit the <A HREF="http://www.Web3Dbooks.com/">XHTML Example by
Example Web site</a> to download source code and resources
related to this book. Visit the
<A HREF="http://www.corexhtml.com/">Core XHTML Web site</A> to learn
more about this companion book to XHTML Example by Example.</P>

</BODY>
</HTML>

**Example 1–2    Sample XHTML Code**

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

  <head>
    <title>Sample XHTML Code</title>
  </head>

  <body bgcolor="white">

    <p><img src="images/logo.png" alt="logo" width="128" height="128" /></p>

    <h1>XHTML Example by Example</h1>

    <p>by <a href="http://www.mantiscorp.com/people/aew/">Aaron E.
      Walsh</a> and <a href="http://www.w3.org/People/Raggett/">
      Dave Raggett</a></p>

    <p><strong>Extensible Hypertext Markup Language</strong>, better
      known as XHTML, is among the latest and greatest technologies
      to hit the Web. Developed by the <a
      href="http://www.w3.org/"> World Wide Web Consortium</a> (W3C)
      as the official successor to the tried-and-true but aging
      Hypertext Markup Language (HTML), XHTML marries the strength
      of HTML's formatting and presentation capabilities with the
      red-hot Extensible Markup Language (XML) that is today
      revolutionizing Internet data representation and information
      exchange. In short, XHTML is a reformulation of HTML in terms
      of an XML language that promises to bridge the wide gap
      between yesterday's HTML-based eye candy Web pages and
```

```
    tomorrow's industrial-strength Web applications.</p>

  <p>In this book we'll start to unravel the mysterious and
     complicated puzzle of Web application development. With an
     emphasis on XHTML, we'll see how critical technologies such
     as HTML, XHTML, and XML come together--along with several
     other W3C technologies--to create sophisticated Web
     applications that we could only dream about a few years ago.</p>

  <p>Visit the <a href="http://www.Web3Dbooks.com/">XHTML Example by
     Example Web site</a> to download source code and resources
     related to this book. Visit the <a
     href="http://www.corexhtml.com/">Core XHTML Web site</a> to
     learn more about this companion book to XHTML Example by Example.</p>

 </body>
</html>
```
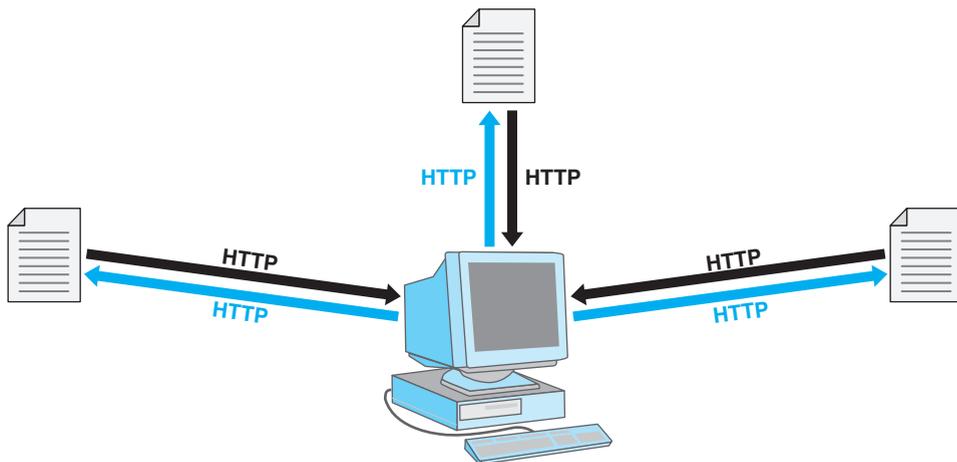
▼ **HTTP (Hypertext Transfer Protocol)** is the formal set of information exchange rules (the *protocol*) that Web clients (browsers) and servers adhere to when talking with each other. Web browsers connect to and exchange information with Web servers via HTTP. A typical HTTP "conversation" revolves around the client browser requesting Web pages (HTML documents) that the Web server sends back, or *serves*, to the browser (forming the basis of a simple client-server relationship), as illustrated in Figure 1–3. Built on



**FIGURE 1–3**    Hypertext Transfer Protocol (HTTP) is the primary protocol used to exchange information on the Web. Typically, all communications between Web browsers (clients) and Web servers take place entirely in the HTTP protocol. The relatively simple client-server nature of the Web fueled numerous browser and server development efforts in the early years.

top of the Internet's lower-level TCP/IP protocol, HTTP is a relatively simple client-server protocol. Browser's today support HTTP and a variety of other data transfer schemes, including FTP (ftp://), NEWS (news:), MAILTO (mailto:), and many more (see Figure 1–4).

| HTTP |
| TCP |
| IP |

**FIGURE 1–4**　HTTP is a relatively simple transport protocol designed to shuttle Web pages across the Internet. Conceptually, HTTP layers on top of the Internet's lower-level Transmission Control Protocol (TCP) and Internet Protocol (IP), together known as TCP/IP. TCP/IP forms the basis of Internet communications and predates HTTP by more than two decades.

Aside from acronym overload, do you notice a common theme here? In a word, the key foundation technologies on which the original Web was built, and atop of which it sits even today, were *simple*. Simple. Simple. Simple. Even when they were first introduced to the Internet community, HTML, URLs, and HTTP were relatively simple technologies inspired by more complex solutions.

HTML 1.0, for example, was a bare-boned application of SGML designed specifically for writing text-only Web pages. (Version 1.0 of the language dealt only with text, and had no concept of images, sounds, applets, or other forms of Web content that we take for granted today!) Similarly, HTTP is a simple but effective transport protocol designed to shuttle Web pages over the Internet (HTTP builds on top of TCP/IP with two low-level Internet protocols, as illustrated in Figure 1–4). And URLs are a simple form of URI that even non-techies can understand, read, and write.

Because the early Web was founded on simple technologies, creating content and tools for this budding information infrastructure was also easy. Software developers created first generation Web browsers and servers with relative ease, while scores of non-programmers quickly found that they could create HTML Web pages without breaking a sweat. And create they did; the Web grew by leaps and bounds because it was a simple, effective information system without boundaries. You didn't have to have a Ph.D. in Computer Science to create Web software products, nor did you have to be a programmer to create Web pages; simplicity kicked opened the floodgates and into the ether content flowed.

Today, the astonishing growth of the Web is now legendary, something we practically take for granted. But a funny thing happened on the way to Web ubiquity. As the Web became more popular, increasing pressure was put on the foundation technologies to support increasingly sophisticated content and capabilities.

HTML, in particular, has grown by leaps and bounds since it was first conceived so many years ago in direct response to demand from software and content developers. Over the years HTML has been stretched, pushed, and pulled to the very limits of its capabilities, setting the stage for more sophisticated markup languages. More specifically, the inherent limitations of HTML led the W3C to develop XML and XHTML.

## Stretching HTML to Its Limit

When HTML was first introduced to the Internet community about a decade ago, it was a very simple markup language based on SGML (technically speaking, HTML is an *application* of SGML, as the next chapter explains in detail). Developed over three decades ago, SGML is a *metalanguage* designed specifically for the publishing industry (just as XML is a metalanguage for the Web publishing industry, XML is an application of SGML created specifically for the Internet).

As a metalanguage, SGML is a specification from which markup languages can be created; SGML gives information publishers a standard, well defined mechanism for constructing application-specific markup language rules (SGML became International Standard ISO 8879 in 1986). In other words, HTML is a markup language for the Web that was derived from the SGML standard.

Like all markup languages, HTML is used to "mark up" pieces of text with special identifiers, or *tags*, as seen earlier in Example 1–1. XHTML is no different, except that every tag must be in all lowercase characters, whereas case isn't an issue for HTML. Consider, for instance, the following snippet of XHTML code from Example 1–2:

```
<p>In this book we'll start to unravel the mysterious and complicated
puzzle of Web application development. With an emphasis on XHTML,
we'll see how critical technologies such as HTML, XHTML, and XML
come together--along with several other W3C technologies--to
create sophisticated Web applications that we could only dream
about a few years ago.</p>
```

Here, the entire block of text is marked up, in this case with the `<p>` and `</p>` tags. The opening `<p>` tag tells your browser, in essence, to format every character of text that follows as part of a *paragraph*, while the closing `</p>` tag tells your browser that the end of the paragraph has been reached. Similarly, the following line of code uses the `<b>` and `</b>` tags to tell your browser to format the word "dickory" in *bold*, while the text "this link" has been marked up as a hyperlink using the `<a>` and `</a>` *anchor* tags:

```
Hickory, <b>dickory</b>, dock, the mouse ran up the clock. The
clock struck twelve. Click <a href="page2.html">this link</a> to
continue.
```

With HTML 1.0, Web browsers simply scanned through Web pages looking for tags defined by that version of the language, presenting the document to the end user as dictated by the tags. Because HTML 1.0 was such a simple language, parsing Web pages written in it was particularly easy. By any account, HTML 1.0 was a lean markup language, and terribly stingy when it came to features and capabilities; HTML 1.0 supported little more than the ability to assemble and format text into hyperlink-enabled documents that lived on the Web.

HTML 1.0 was enough to get the Web page publishing ball rolling, but it didn't roll very fast or very far before content developers and browser developers grew impatient. A major update to the language was clearly needed, a job that was spearheaded by the Internet Engineering Task Force (IETF). IETF published the HTML 2.0 specification in 1994, the same year that Netscape Navigator 1.1 (a commercial browser developed largely by the same crew that earlier created the NCSA Mosaic browser widely viewed as the first killer Web application) was released. That same year the W3C was formed to shepherd future development of HTML and related Web standards.

As the HTML standard evolved through the W3C, browser vendors aggressively sought to differentiate their products with their very own custom tags. Because HTML wasn't flexible, it could not be gracefully extended as XHTML can be today. At the time, browsers had to be continuously revised in order to keep up with the influx of new tags, regardless of whether the tags were proprietary or standard. In the end, everyone lost.

Because HTML wasn't easily extended, Web browser developers became locked in vicious upgrade cycles as they struggled to outdo each other in terms of standard and non-standard HTML feature support. Although this rapid evolution is in its own way responsible for advancing the state of the art on the Web, in many cases content developers simply couldn't keep up with the pace of change, nor could they create pages that displayed consistently across all browsers. End users became frustrated and disenchanted during the height of the "browser wars" because there was simply no way to keep up with the firefight. And, perhaps worst of all, HTML suffered greatly as the responsibility of the language shifted to the newly established W3C, which couldn't immediately resolve the incompatible demands of its members and the Web community at large.

**Online**

At the time of this writing, Internet Explorer is used by approximately 90% of Web surfers. Netscape 6 was recently released, and promises to bring a much needed injection of standards-compliance to the field as the sun sets on the browser wars. Visit http://www.mantiscorp.com/people/aew/snippets/ to learn more about how the browser wars stretched HTML to its limit, ultimately leading us to XHTML.

Although HTML 4.0 goes a long way toward solving the problem, as it offers the richest suite of standard HTML features to date, the fundamental problem remains: HTML isn't easily extended or customized. Regardless of how feature-packed HTML 4.0 is, Web developers will always want more. Aside from introducing proprietary tags, browser vendors have no way of extending or enhancing HTML. This inherent limitation of HTML is one of the primary reasons that XHTML was conceived, but it's not the only reason. Not by a long shot.

HTML 4.0 was released by the W3C in 1997, followed more recently by HTML 4.01. Aside from small fixes to its predecessor, HTML 4.01 is significant in that it is the base language on which XHTML is built (you'll often hear that XHML is based on HTML 4.0, but in truth it's actually based on HTML 4.01). In 1999 the W3C began to recast HTML 4.01 in terms of XML, giving us XHTML. Without question, XHTML is essential because:

▼ HTML isn't easily extended or customized

▼ HTML is inadequate for automated data exchange

▼ HTML does not play well with other markup languages

▼ HTML parsing is often ambiguous

**Warning**

In the next chapter we'll dig deeply into the main reasons why XHTML was developed.

## XHTML Testimonials

If you're not already convinced that XHTML is the future of Web page development, consider what the experts are saying. Following are just a few of the public statements made by professional Web developers on behalf of XHTML 1.0 when it debuted, many of which emphasize the importance of its backward compatibility with HTML:

*As a leading supporter of XML and HTML 4.0,* **Netscape is pleased by the re-
lease of the XHTML 1.0 Recommendation***. XHTML provides a clean, fully
XML-compliant syntax for the large amount of content already available on the
Web and will make this content accessible to tools and applications that support
XML. The Netscape/Mozilla browser currently in progress will provide full HTML
4.0 compliance, making it a short, easy transition to XHTML 1.0 in a subsequent
release. Content written in HTML 4.0 will migrate easily to XHTML****. Together
with other W3C standards like CSS, DOM, and XML, XHTML will provide
greater structure, power, interoperability, and control for documents and
data on the Internet***.*

> *—Jim Hamerly, Vice President of Client Products, Netscape Communications
> Corporation, a subsidiary of America OnLine, Inc.*

*XHTML is an important open standard that will help our e-business customers de-
ploy to web applications on any browser via XML-enabled servers.* **IBM is excited
by the power that XHTML will bring to pervasive computing** *as the need to
deliver information to devices such as cell phones, PDAs, and set top boxes grows
exponentially.*

> *—Marie Wieck, Director of XML Technology, IBM Corp.*

**HP is thrilled about XHTML** *1.0 becoming a W3C Recommendation. Our HP
ChaiFarer Web browser supports XHTML for embedded systems, thereby allowing
easy customization of the Web experience for different classes of devices.*

> *—William Woo, General Manager, Embedded Software, Hewlett-Packard*

**Ericsson is very happy to see XHTML 1.0 made a W3C Recommendation***.
We are now able to see a realistic and rapid path for the convergence of the mobile
and fixed-network Web. Mobile devices already use an XML-based markup lan-
guage (WML), and convergence of WML towards XHTML, as well as use of data
from the Web on mobile devices, is tremendously simplified by the W3C formaliza-
tion of the XHTML specification. We have been working hard in both the W3C and
the WAP Forum to enable this, and while a lot of work is yet to be done,* **this is a
big step forward.**

> *—Kari Laihonen, Manager, IT Standardization, Corporate Technology, Ericsson*

*The Web3D Consortium is a non-profit organization dedicated to the development
and support of open standards for 3D graphics on the Web. We enthusiastically
support XHTML advancement as a W3C Recommendation. Our concurrent efforts
producing the Extensible 3D (X3D) Graphics specification benefit directly from
XHTML progress.* **Authors will be able to produce rich multimedia content
that includes X3D scenes, SVG, MathML, audio, video, and streaming
events, all integrated compatibly within an XHTML hypermedia frame-
work***. We will continue working with W3C activities to produce compatible and
interoperable technologies.*

> *—Don Brutzman, Vice President of Technology, Web3D Consortium*

*Extensibility, modularity, interoperability—the XHTML 1.0 specification brings together the needs of the Web designer in a tidy XML-based package.* **As the stepping stone between HTML and XML, XHTML allows a web creator to begin using XML today in web applications, while maintaining compatibility with existing HTML browsers***.*

> *—Ann Navarro, Vice-President, HTML Writers Guild*

*XHTML 1.0 acts as a bridge between the first and second generation mark-up languages, and allows for seamless transition to XML use. Due to its modular construction and greater flexibility,* **XHTML allows developers to create their own tags and add features specific to their purpose. This flexibility is a key component in Web-enabling wireless and handheld devices.** *Ask Jeeves intends to use XHTML 1.0 throughout their site to improve the user experience, enable easier site maintenance, and increase time to market.*

> *—Daniel Austin, Senior Programmer and Web Site Architect, Ask Jeeves*

## Summary

XHTML is a reformulation of HTML in terms of XML, which was brought about to remedy several HTML limitations. XHTML promises to bridge the gap between yesterday's HTML-based Web pages and tomorrow's most sophisticated XML-based Web applications. XHTML 1.0 was released by the World Wide Web Consortium (W3C) in January 2000, and officially makes HTML obsolete. Based on HTML 4.01, XHTML 1.0 brings the rigor of XML to the Web browser. Because it is backward compatible with HTML 4.01, properly constructed XHTML Web pages can be viewed using today's current Web browsers. As a modular XML-based language, XHTML can be integrated with other XML languages to produce rich Web pages, and can also be used to construct Web content for a variety of Web devices, including mobile phones, Personal Digital Assistants (PDAs), wireless handheld computers, and Internet appliances.

### Summary of URLs Found in This Chapter

World Wide Web Consortium (W3C)    **http://www.w3c.org/** or **http://www.w3.org/**

XHTML Example by Example Web site    **http://www.Web3Dbooks.com/**

Core XHTML Web site    **http://www.CoreXHTML.com/**

Aaron E. Walsh Web site    **http://www.mantiscorp.com/people/aew/**

Dave Raggett Web site    **http://www.w3.org/People/Raggett/**

Snippets    **http://www.mantiscorp.com/people/aew/snippets/**