The Web is a powerful front-end for an application, particularly an application built with Oracle PL/SQL.

To build applications on the Web, it is vital to understand what makes the Web work. This chapter reviews the basics of the Internet, the Web, and how a browser works to retrieve a page that you request. The concept of a Web application is explained, both for intranets and for the Internet. Finally, the advantages of using PL/SQL for Web applications are discussed.
THE INTERNET

Ask the next person who says Internet what they mean by the term, and you may get a surprisingly vague reply. Simply put, the Internet is a decentralized, worldwide network of interconnected computers that communicate with each other in a standardized way. The computers are connected using modems and phone lines, cable lines, ISDN lines, Ethernet cards, and fiber-optic cables. To provide convenient wireless communication, infrared and radio waves are being used increasingly as well.

Physical connectivity is not enough to define the Internet. Communication would not be possible if these interconnected computers did not use a shared protocol for transmitting information. A protocol is a set of common standards and software for communication. You can think of it as a common language utilized for communication between all of the different types of computers and operating systems across the Internet. If an English-speaking person in London calls a French-speaking person in Paris, the telephone network connects the two parties, but that does not guarantee that they understand each other. To send information such as an email from one computer to another, both computers, as well as any intermediate computers the message must travel through, must be able to read instructions about the type of message, its source and destination, and how it should be handled, so that it can be successfully transmitted to and read by the intended recipient.

The protocol of the Internet is TCP/IP, which stands for Transmission Control Protocol/Internet Protocol. TCP/IP actually consists of two protocols working together, TCP and IP, though they are commonly referred to as a single protocol, TCP/IP. Messages sent according to this protocol contain a header—beginning lines of text that are separate from the information being sent. This header provides instructions for transmitting the data. In ground mail, letters are enclosed in envelopes that contain addresses in a standard format, as well as conventional phrases for special handling such as “Air Mail,” “Poste Restante,” or in less fortunate cases, “Return to Sender.” Similarly, when information is sent across the In-
ternet, it is sent with a TCP/IP “envelope” that indicates who originated the transmission, who should receive it, and other instructions to ensure the information is sent and received correctly.

To make it possible to find a machine on the Internet, each Internet location has been assigned its own address, called an IP address, or Internet Protocol address. The IP address is used in TCP/IP when specifying the source and destination for the transmission. An IP address is a 32-bit number. Rather than using a cumbersome 32-digit procession of ones and zeros, an IP address is expressed as four numbers separated by decimal points. An example of an IP is 192.239.92.40. Each number can range from 0 to 255, resulting in 256 possibilities, or 8 bits (2 to the power of 8), so each number can be referred to as an “octet.”

Sometimes information being transmitted via TCP/IP has its own internal protocol as well. For example, Web pages are transmitted via TCP/IP, but because they are Web pages, they are also sent with an additional header that is compliant with the protocol HTTP, or Hypertext Transfer Protocol. This protocol, HTTP, is a familiar beginning, or prefix, for Web page addresses.

WHAT IS THE WEB?

The World Wide Web, the Web, or the W3, was started by scientists working at CERN, the European Organization for Nuclear Research, in Bern, Switzerland. They needed a convenient way to collaborate and share technical information with their colleagues. By publishing documents on the Internet, they could access colleagues’ technical papers and related information easily. Tim Berners-Lee is credited with being the “Father of the Web” for his vision on how to accomplish this. Standards for the Web are maintained currently by the W3C, or World Wide Web Consortium, an international organization. Its Web site is http://www.w3.org.

As explained earlier in this chapter, there is a separate protocol, HTTP, designed specifically for handling Web pages. Web pages themselves are constructed primarily in HTML. HTML, or Hypertext Markup Language, started as a way to add formatting commands, or “markup” in publishing terms, to a technical paper. The “hypertext” part of the name refers to text that can be selected to provide a link to another document. This makes information retrieval easy, and it is one of the hallmarks of the Web; for example, a technical paper that refers to previously published results can include a link to that previous paper, and the user can click back and forth between documents. HTML is covered in further detail in Chapter 5, “Introduction to HTML: Basic Tags, Tables, and Frames,” and Chapter 6, “Advanced HTML: Forms, Nested Tables, and Nested Frames.”

For a preview of what HTML looks like, go to a Web page, right-click in the browser window, and choose “View Source.” This text is what your browser is reading and displaying in the browser window.
Usage of the Web quickly spread beyond the scientific community. With the advent of browsers such as Netscape and Internet Explorer, many users could easily enjoy the ease of publishing and accessing material on the Internet. The Web was born from a need to exchange ideas freely, and that spirit is a large part of the success of the Web.

**HOW DO WEB BROWSERS WORK?**

The Web browser’s basic role is to take a request for a Web page from a user, and then display that page to the user. This is referred to as “rendering” a Web page.

Typing the address of a Web page into a browser to retrieve a Web page is an increasingly familiar activity. What happens behind the scenes?

When a user requests a Web page, the browser locates the Web page on the Internet, retrieves it, and displays it in the browser’s window.

Someone who is enthused about U.S. tax returns might enter the following address:


This address is also known as a URL. A URL, or Uniform Resource Locator, is the complete address for locating a specific page, or “resource,” on the Internet. Each Web page has a unique URL, and to view a Web page, you must know its URL. You can also use the term URI, Uniform Resource Identifier, though this is really synonymous with URL.

The first step for the browser is to identify who has the file being requested by the user, using the URL. The browser divides a URL such as the one displayed above into three parts. The first part of the URL is the protocol. In this case, the protocol is “http://”, which indicates that the file should be retrieved using HTTP, or Hypertext Transfer Protocol. Another protocol you may see often, particularly if you conduct any business on the Web, is HTTPS. This indicates that a Secure Socket Layer, or SSL, is layered on top of the HTTP protocol for additional security. Despite the HTTP or HTTPS, realize that communication also still takes place using the protocol TCP/IP. TCP/IP governs the connection and transmission of the information, whereas HTTP is concerned with the Web page itself. Because the protocol is identified as HTTP in the URL, the browser knows how to structure its request for the Web page.

The second part of the URL is the domain. In this example, it is “www.irs.gov”. The domain identifies a specific server machine on the Internet. A domain ends with characters that give a general indication of the site’s purpose, such as .com, .org, .net, .gov, .mil, and .edu, as well as many country suffixes such as .uk for United Kingdom, .de for Germany (Deutschland), and .fi for Finland. The middle portion of the domain provides specific information on the site being accessed. In “www.irs.gov”, it is “irs”, which happens to be the United States Internal Revenue Service, or IRS. For name recognition, most organizations use their own
name, or a familiar abbreviation of it, for this part of a URL, unless the preferred URL is already claimed by another organization.

The leftmost part of the domain is the actual machine name, or hostname. Many organizations use “www” as a convention, instead of a specific machine on their network. When a request arrives for a page on the “www” machine, the company’s network has configuration files that redirect such requests to an appropriate machine. Using “www” instead of a specific machine name makes it easier for a company to move pages from one machine to another, or to use multiple machines to handle requests, because the user is not specifying a machine name.

As discussed earlier in this chapter, machines on the Internet are identified by an IP address—four numbers divided by decimal points, such as 192.239.92.40. Yet the domain listed above, “www.irs.gov”, is alphanumeric. Which is the correct syntax to identify a machine on the Internet? In fact, both methods of identifying a machine on the Internet are correct. One advantage of using a name like “www.irs.gov” is that it is easier to remember than a numeric IP address, to the relief of U.S. taxpayers on April 14. Another advantage is that the systems administrator can associate an alphanumeric domain with another IP address if need be. If you use the IP address for a site that has been moved to a new location on the Internet, you may have problems getting to the site.

If you know the IP address of the site you wish to visit, then you can use that IP address in place of the domain name, and a browser can understand the request. In fact, you eliminate an extra step for the browser. When an alphanumeric domain is used, the browser has to translate that domain name into a numeric IP address before it can request the Web page. To resolve domains, the browser contacts a specific type of server known as a Domain Name System server, or DNS server. The browser may need to contact several DNS servers to translate the domain into an IP address. Companies may also have an internal DNS server, which must be contacted to provide further translation of domains within a company. Eventually, the browser resolves the domain name into a valid IP address, or returns an error.

The third part of the URL, to the right of the domain, is optional. It may be a file name, perhaps with a path. This part refers to the specific resource being requested. In this example, it is “/search/srchelp.html”. The user is requesting the resource called “srchelp.html”, which is located in a directory called “search.” If the URL does not contain this third part, then a default page is displayed. Most Web sites have an initial page with an address that contains only the domain name, for ease in remembering the URL.

Once the browser reads the URL and knows the protocol to use, the IP address of the machine to contact, and the specific path and file being requested, if any, it initiates a request. It connects to the IP address using a specific point of connection on the target server called a port. Ports are numbered, and by convention, the overwhelming majority of systems administrators configure their machines to use port 80 as the default for HTTP Web documents. For this reason, the port number is not usually included in a URL, but the port can be included after the
domain if needed. The syntax is a colon followed by the port number. The IRS example could also have been written:


to retrieve the same Web page as:


Through the connection it has established, the browser sends a request, called a GET request, for the Web page. The server processes the request by looking for the directory and file requested. Sometimes the request is for a page that exists on the server already; this type of page is a static page, and the server can transmit the page to the browser that requested it. In other cases, the page has to be generated. The URL invokes scripts that produce the page dynamically, and then transmit the page. This is how you create the application in this book—using PL/SQL to dynamically generate Web pages that contain database content.

If the request is successful, the server returns the requested page to the browser. If it cannot find the resource being requested, it returns an error message to the browser.

Although a user apparently requests a single file to view a Web page, that page may in turn spark additional requests for files to complete the Web page. When rendering a Web page, the browser may find that the page contains several images, which need to be retrieved in turn. Images on a Web page are themselves separate files, as are audio and video files. If you request a Web page with six separate images, the browser will need to retrieve seven documents to display the full page. Thankfully, the browser figures out which additional files are needed, requests them one by one, and assembles them for you, which would be a tedious task otherwise.

This transparent operation allows users to think of visiting a single Web page, even though that one page may be a combination of several files, or the result of a script that created the page upon request.

**Unit 1.1 Exercises**

a) What is the protocol of the Internet?

b) What is the main protocol of the Web?
UNIT 1.1 EXERCISE ANSWERS

a) What is the protocol of the Internet?
   Answer: TCP/IP, which contains instructions for how to handle a message being transmitted from one location to another across the Internet.

b) What is the main protocol of the Web?
   Answer: HTTP. If a message contained within a TCP/IP transmission has an HTTP header, this indicates that the message happens to be a Web transmission; for example, a Web page or a request for a Web page.

c) Will the two addresses below retrieve different pages? Why or why not?
   http://www.bbc.co.uk:80
   http://www.bbc.co.uk

   Answer: They retrieve the same page. The default port for HTTP is 80, so including the port number 80 in the URL is not necessary.
UNIT 1.2

WHAT ARE WEB APPLICATIONS?

ADVANTAGES OF WEB APPLICATIONS

Applications are software designed to meet a specified need. Word processing software, spreadsheet programs, and data entry tools are some examples of applications.

A Web application is just an application that is deployed on the Web. It is a Web page, or series of Web pages, allowing users to accomplish a task like obtaining information and forms, shopping, applying for a job, listening to Internet radio, or any of the many activities possible on the Web. To use a Web application, a user needs to know a URL for the application, and possibly a name and password. Another way to think of a Web application is a Web site offering a great deal of functionality.

Web pages are truly powerful when they are connected to a database. Such pages are referred to as database-driven pages. Pages are generated dynamically, and include up-to-the-minute information, without requiring someone to manually update the content of the Web page. Pages can provide current news, stock quote information, product availability data, or shipping information by retrieving information from a database at the time pages are requested. Users in turn conduct transactions directly with the database in real time. Connecting a Web page to a database truly makes for a powerful Web site. It is said that Web site programmers eventually find that they need to connect their site to a database, while database developers find that sooner or later their database needs to be accessible through the Web.

Businesses with a Web presence tend to be viewed as more competitive in the marketplace, and some conduct business entirely on the Web. Deploying an application on the Web gives an organization a way to extend their reach beyond the borders of the organization to the public directly. A Web site can be made publicly accessible to anyone with an Internet connection. Interactions with cus-
customers that were previously handled by phone, by mail, or in person can be handled via the Web. For example, consumers are familiar with the concept of an online retail store that lets users shop for goods and services, fill up a shopping cart, and “check out” their purchases by paying with a credit card.

INTERNET VS. INTRANET

Organizations can set up an internal network, known as an intranet. An intranet protects the organization from having its internal files and confidential information accessed by someone outside the company. Intranets are often used to facilitate file sharing and email among members of an organization, while blocking access to outsiders.

Web applications can be deployed on an intranet, rather than the Internet. This simply means they are accessible to a smaller group of individuals. Web applications for internal use can be restricted to specific users, and in some cases, specific IP addresses.

When companies create custom applications tailored to meet their internal needs, they make a decision about what type of “front-end” to create, or what interface the users will work with when they use the application. Internal Web applications, restricted to specific users or computers, are increasingly popular within companies. Enterprise applications, which are larger applications used by every area of a company with the aim of every employee conducting business on the same, integrated system, are also being designed (or redesigned) for the Web.

Consider the advantages of using the Web from the company’s perspective. An application that is not deployed on the Web typically requires employees to have some type of system software installed on their machines. Installation can be a time-consuming process that grows more burdensome as new versions of the software are released and must be installed on each computer. For example, an annoying screen problem is fixed in Version 1.3.5.2 of the software, but you only have Version 1.3.5.1, so you need to install the update to fix the problem.

Monitoring is needed as people leave and join the company and as computers are reused. This is in sharp contrast to an internal Web site, which only requires a Web browser and an Internet connection. Web browsers come preinstalled on the majority of office computers, and many companies connect computers to networks as a matter of course. In Web architecture, which is discussed further in Chapter 2, application code is housed on server machines instead of on each computer. New versions of Web applications can be made available immediately to all users at the same time by making changes on the servers, instead of going to each individual user’s machine.

Perhaps the main advantage of choosing a Web interface is that many employees are already familiar with it. The conventions of browsing the Web—clicking on links, navigating pages, entering data on forms—are fairly consistent. More people are using the Web for shopping or to retrieve information. Colleges and uni-
Universities are producing more graduates accustomed to having sophisticated online services to register for classes, check their accounts, request transcripts, and update their personal information. When new applications require less training because they have a familiar look and feel, then a company can save a considerable amount of money in training.

Employees may also more readily accept a new application if the interface is a familiar and intuitive one, ensuring that a company’s investment in software is a worthwhile one. An elegant and powerful system is neither if it is never used.

The ease of providing information to their employees on the Web has made companies increasingly think of ways to computerize and Web-enable their internal operations. Many organizations have an internal Web “home” site, restricted to employees only, for publishing relevant information and links. Internal sites are more than a way to communicate information. As a company computerizes more of its operations, an employee can conduct day-to-day business quickly and easily on the Web: request meeting rooms, fill out timesheets, request travel reimbursements, apply for internal job postings, and access up-to-date employee directories, to name a few possibilities. The relative ease of developing for the Web means that more and more business functions can be automated on-line. Although the “paperless office” is more of a dream than a reality, paperwork or processing time often can be minimized or eliminated by moving to a Web-based application.

**ADVANTAGES OF PL/SQL FOR WEB DEVELOPMENT**

SQL is the industry-standard language for reading, inserting, updating, and deleting data in a database, so it is an essential part of creating dynamic Web pages. PL/SQL, or Procedural Language/Structured Query Language, is Oracle’s extension to SQL. PL/SQL adds the sophisticated control structures of a procedural language, such as loops, if-then-else statements, variable declaration, and exception handling. It was first introduced by Oracle in 1988 and has continued to mature since its introduction. Oracle is committed to supporting PL/SQL, and regularly produces enhancements and refinements to the language.

Because PL/SQL is compiled and stored directly in the database, it has the distinct advantage of being fast. It does not require compilation or interpretation at runtime, and it interacts with the database quickly. Other existing technologies that generate Web content by making use of a database require many trips to and from the database. A JSP, ASP, or Perl script run on the server might go back and forth to the database while generating the HTML page. PL/SQL has the advantage of being in the database, so none of these back-and-forth trips are required.

The advantages in speed are particularly apparent as the volume of data being handled increases. The code is also reliable and secure. Backup and recovery processes for the database also save the PL/SQL code stored in the database. The security and access controls put in place for Oracle database objects, such as tables, also protect PL/SQL packages, procedures, and functions, which are themselves database objects.
As the title of this book promises, the PL/SQL language that is so effective at handling back-end processing can also generate the front-end application that the user sees. Oracle’s 9i Application Server, previously known as the Oracle Application Server (OAS), is the middleware that enables the PL/SQL stored in the database to generate Web pages displayed in a browser. System architecture, middleware, and the Oracle 9iAS are described in more detail in Chapter 2.

Companies using the Oracle 9i database already have PL/SQL, since it is included with the database and does not require separate licensing. Companies using Oracle are also likely to have PL/SQL expertise in-house; there is a large knowledge base of PL/SQL programmers.

**UNIT 1.2 EXERCISES**

a) What is the main factor that determines whether an application will be deployed on an intranet or the Internet?

b) Why is a Web interface becoming the accepted standard for applications?

c) Does a PL/SQL programmer work on an application’s front-end or back-end?

d) Describe some advantages of using PL/SQL for Web applications.

**UNIT 1.2 EXERCISE ANSWERS**

a) What is the main factor that determines whether an application will be deployed on an intranet or the Internet?

*Answer:* The audience for the application determines where it will be deployed. If you want to limit access to users within an organization, use an intranet.

b) Why is a Web interface becoming the accepted standard for applications?

*Answer:* Web applications are relatively easy to build and deploy through leveraging existing skill sets within the business. Users are familiar with the interface, and only need a browser, which is installed on most computers already.
c) Does a PL/SQL programmer work on an application’s front-end or back-end?
   Answer: Both. A PL/SQL programmer can write programs that manipulate data within the database, behind the scenes. A PL/SQL programmer can also write programs that generate dynamic Web pages for the user.

d) Describe some advantages of using PL/SQL for Web applications.
   Answer:

   Speed—PL/SQL is stored in the database, so multiple back-and-forth trips to the database are avoided.

   Security—PL/SQL programs are database objects and are protected by Oracle’s database security.

   Integration—You can use Oracle for the entire application, from front-end to back-end.

   Flexibility—You can apply existing PL/SQL skills to front-end work or back-end data processing.