

WEB SITE DEVELOPMENT

A-TO-Z FOR BEGINNERS

Version 3.01.01.08.27

Edward B. Toupin

All trademarks mentioned throughout this publication are property of their respective owners.

Electronic Publishing by
Online Books Inc., PO Box 268, Mt. Vernon, VA 22121
<http://www.onlinebooksinc.com>, editor@onlinebooksinc.com.

Copyright © 2001, Edward B. Toupin. All rights reserved.

Manufactured in the United States of America.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, recording, or otherwise without the prior written permission of Edward B. Toupin.

Provide your feedback at <http://www.delphi.com/onlinebooksinc/>.

Table of Contents

List of Figures	x
List of Listings	xii
List of Tables	xiv
List of Examples	xvi
About The Author	xviii
Acknowledgements	xx
Preface.....	xxii
1. Introduction	1
2. Web Site Overview	3
2.1 What is the Web?	3
2.2 How does it work?	3
<i>Presentation Layer</i>	4
<i>Application Layer</i>	5
<i>Comm Layer</i>	5
2.3 Web Site Software	6
<i>The Web Server</i>	6
<i>The Web Client</i>	7
2.4 Obtaining a Web Site	8
<i>Virtual Hosting</i>	8
<i>Dedicated Server</i>	10
<i>Sub-Domains</i>	11
<i>Free Space</i>	11
3. Basic HTML.....	13
3.1 What is HTML?	13
<i>Hypertext and Markups</i>	13
<i>Tag, You're It!</i>	13
<i>Tag Pairs</i>	14
<i>Attributes</i>	14
<i>Things to Note</i>	14
3.2 What does a basic page look like?	14
3.3 What are the different parts of HTML?	15
<i>Additional HEAD Tags</i>	16
<i>Body Attributes</i>	16
<i>Content Tags</i>	17
<i>Image Tags</i>	17
<i>Hypertext Link Tags</i>	18
3.4 Building a Page	19

4. Advanced HTML	21
4.1 Graphic Links and Image Maps.....	21
<i>Adding Images</i>	21
<i>Adding Imagemaps</i>	21
<i>Things to Consider</i>	22
4.2 Animated Graphics.....	23
<i>Animated GIFs</i>	23
<i>Transparent GIFs</i>	23
4.3 Using Frames.....	24
<i>Framesets and Frames</i>	24
<i>Frameset Example</i>	26
<i>Targeting Links</i>	27
<i>Screen Layout</i>	27
<i>Some Things to Consider</i>	28
4.4 Forms.....	28
<i>The Form Tag</i>	29
<i>Input Type</i>	29
<i>Password</i>	30
<i>Checkbox</i>	30
<i>Radio</i>	31
<i>Hidden</i>	32
<i>Submit and Reset</i>	32
<i>TextArea Input</i>	33
<i>Drop-Down Menus</i>	33
4.5 Managing Form Input	34
<i>E-Mail</i>	34
<i>JavaScript</i>	35
<i>PERL/CGI</i>	35
4.6 Building a Form	35
<i>The Frameset Page (main.htm)</i>	35
<i>The Navigation Page (nav.htm)</i>	36
<i>Form 1 Page (form1.htm)</i>	36
<i>Form 2 Page (form2.htm)</i>	37
<i>How does it work?</i>	38
4.7 Browser Compatibility Issues	38
<i>Problems for Web page Display and Performance</i>	38
<i>Resolving Issues</i>	40
5. JavaScript.....	43
5.1 What is JavaScript?	43
<i>What is Object Oriented Programming?</i>	43
<i>Objects and Properties</i>	44
<i>Methods</i>	44
<i>Functions</i>	44
<i>Events</i>	45
5.2 Adding JavaScript to Web Pages	45
5.3 Compatibility Across Browsers	46
5.4 Writing JavaScript	46
<i>Methods and Properties</i>	46
<i>Events</i>	47

<i>Writing Your First Script</i>	48
5.5 Other Applications	49
<i>Dynamically Change a Document's Background Color</i>	49
<i>Status Bar Messages</i>	49
<i>Dynamic Text Manipulation</i>	50
<i>JavaScript Dialog Boxes</i>	50
<i>Image Submit Button</i>	51
<i>Displaying a Random Message</i>	51
<i>Live JavaScript Clock</i>	52
<i>Image Flip</i>	53
<i>Drop-Down Menu Box</i>	53
<i>Other Places to Look</i>	54
6. CGI Programming	55
6.1 What is CGI Programming?	55
<i>Reading the User's Form Input</i>	55
<i>Sending the Response Back to the User</i>	55
6.2 Adding CGI to a Web Page	56
<i>Links</i>	56
<i>Server-Side Includes</i>	56
<i>Form Actions</i>	57
6.3 Locating CGI Scripts	58
6.4 Installing CGI on a Web Server	58
<i>First Stop - Your Host</i>	59
<i>Modifying Your PERL File</i>	59
<i>Understanding File Permissions</i>	60
<i>Uploading And Setting Permissions</i>	60
6.5 Writing a CGI Script	61
<i>The Program</i>	61
<i>The Web Page</i>	61
6.6 Integrating a Database	62
<i>Database Server</i>	62
<i>Flat File</i>	63
<i>Other Mechanisms</i>	63
6.7 Other Applications	63
<i>Cross Browser Compatibility</i>	64
7. ASP Programming	65
7.1 What is ASP Programming?	65
<i>ASP Pages</i>	65
7.2 ActiveX Controls	66
<i>Objects</i>	66
<i>Interfaces</i>	67
<i>Using ActiveX Controls</i>	67
7.3 Adding ASP to a Web Page	68
7.4 Locating ASP Scripts	69
7.5 Installing ASP on a Web Server	69
<i>Windows NT 4.0 Option Pack</i>	69
<i>Windows 98 Install Disk</i>	69
<i>ASP for Other Servers</i>	70

7.6 Writing ASP Scripts	70
<i>Detecting a Browser</i>	70
<i>Another Example</i>	71
7.7 Integrating a Database	72
<i>Another Example</i>	72
7.8 Other Applications	73
8. Java Applets	75
8.1 What is Java?	75
8.2 What are Java Applets?	75
8.3 Adding a Java Applet to a Web Page	75
<i>Installing an Applet</i>	76
<i>Troubleshooting Your Applet</i>	76
8.4 Locating Java Applets	77
8.5 Other Applications	77
9. Advanced Programming	79
9.1 Cold Fusion	79
<i>Application Server</i>	79
<i>What do I need?</i>	79
<i>Selecting Data and Generating Output</i>	80
<i>What's Next?</i>	81
9.2 Dynamic HTML	81
<i>Document Object Model (DOM)</i>	82
<i>Cascading Style Sheets</i>	82
<i>Scripting</i>	83
<i>What next?</i>	83
9.3 Java Server Pages	83
9.4 PERL Hypertext Pages (PHP)	84
<i>A PHP Page</i>	84
<i>Dealing with Forms</i>	85
<i>What's next?</i>	85
10. Remotely Hosted Applications	87
10.1 What are Remotely Hosted Applications?	87
10.2 Adding Remotely Hosted Applications to a Web Page	87
10.3 Locating Remotely Hosted Applications	87
10.4 Other Applications	87
11. Putting It All Together	89
11.1 Putting a Web Site on the Internet	89
<i>Verify Paths</i>	89
<i>Copying the Files</i>	89
<i>CGI Scripts</i>	89
<i>Java</i>	90
<i>Setting Up Databases</i>	90
<i>Application Servers</i>	90
<i>File Permissions</i>	90
11.2 Obtaining a Domain Name	90
<i>Choosing a Domain Name</i>	91

<i>Dot Com?</i>	91
<i>Registering a Domain Name</i>	91
A1. General Links	95
A2. Research Links	97
A3. Tools and Software	99
A4. Web Site Marketing Plan	101
Index	103

List of Figures

Figure 1: The World Wide Web	3
Figure 2: Web Communications	4
Figure 3: Internet Server Ports	6
Figure 4: Web Server	7
Figure 5: Web Client.....	8
Figure 6: Virtual Hosting	9
Figure 7: My First Page	20
Figure 8: Standard Frames.....	26
Figure 9: Nested Frames	28
Figure 10: My Form Frames	38
Figure 11: CGI Call.....	55
Figure 12: SSI Call	56
Figure 13: Integrating a Database	62
Figure 14: Using a Flat File.....	63
Figure 15: Interfaces.....	67
Figure 16: ODBC Data Sources	80
Figure 17: DHTML Hierarchy.....	83

List of Listings

Listing 1: Tags	14
Listing 2: Tag Attributes	14
Listing 3: Sample HTML Page	15
Listing 4: Additional Head Tags	16
Listing 5: My First HTML File	19
Listing 6: Imagemap Code	22
Listing 7: NoFrames Code	26
Listing 8: Sample Frameset Code	27
Listing 9: Nested Frames	28
Listing 10: Checkbox Form Code	30
Listing 11: Radio Button Form Code	31
Listing 12: Submit and Reset Buttons	32
Listing 13: TextArea Code	33
Listing 14: Drop-Down Menu Code	33
Listing 15: Sending Form Input via E-mail	35
Listing 16: Main Frameset Page Code (main.htm)	36
Listing 17: Navigation Page Code (nav.htm)	36
Listing 18: Form 1 Page Code (form1.htm)	36
Listing 19: Form 2 Page Code (form2.htm)	37
Listing 20: Sample JavaScript Enhanced Page	45
Listing 21: Pop-Up JavaScript Window	48
Listing 22: Live JavaScript Clock	52
Listing 23: Sample CGI Form	58
Listing 24: Sample ASP Script Using an ActiveX Control	67
Listing 25: Detecting a Browser with ASP	70
Listing 26: Detecting Time-of-Day with ASP	71
Listing 27: Collecting User Information with ASP	73
Listing 28: Selecting Data and Generating Output with Cold Fusion	80
Listing 29: Sample Java Server Page (JSP)	83

List of Tables

Table 1: Basic HTML Tags	15
Table 2: Body Attributes	16
Table 3: Content Tags	17
Table 4: Image Tags.....	18
Table 5: Hypertext Link Tags.....	18
Table 6: Frameset Attributes	24
Table 7: Frame Attributes	25
Table 8: Form Attributes	29
Table 9: Text Input Attributes.....	29
Table 10: Checkbox Input Attributes.....	31
Table 11: Radio Button Input Attributes.....	32
Table 12: TextArea Input Attributes	33
Table 13: Drop-Down Menu Attributes	34
Table 14: JavaScript Events and Actions	47
Table 15: Various JavaScript Functions	50
Table 16: Permission Settings	60

List of Examples

Example 1: Text Input Form	29
Example 2: Password Input Form.....	30
Example 3: Checkbox Input Form.....	31
Example 4: Radio Button Input Form.....	31
Example 5: Submit and Reset Buttons	32
Example 6: TextArea Input	33
Example 7: Drop-Down Menu.....	34
Example 8: Sample JavaScript Enhanced Form	45

About The Author

Edward B. Toupin is a freelance consultant, writer, and published author living in Las Vegas. He currently handles technical writing tasks for various companies in New York, Chicago, and Denver as well as imagineers and markets feature-length and short screenplays.

Edward provides quality Web site design, development, and marketing as well as writing, document design and planning, and e-book publishing services. You can visit his Web site at <http://www.toupin.com/> or contact him at etoupin@toupin.com.

Acknowledgements

This e-book was developed from the numerous questions about Web development that I received from my clients. Some of the more prominent are summarized as:

- What is it?
- How does it work?
- How can I do it?

I want to thank all of my clients for their feedback, patience, and successes to give me the means to provide this e-book for the masses.

Preface

I've spent hundreds of hours working with clients interested in either developing a new site or enhancing their existing one. In both cases, their primary question has been "Where do I start?" Some clients spent countless hours searching the Web looking for *instructions*, and trying to translate what they found, only to become confused and somewhat overwhelmed with all the information. Of course, they could **buy** a book, but since there's so much free information on the Web, why spend the money?

The purpose of this e-book is to help the novice developer by providing an overview of those topics important to successful Web development endeavors. The information covers everything the new developer needs to know from the basics of HTML through ASP, PERL, and Java programming. Of course, if you're new to the idea of *programming*, these terms are frightening. But, we'll take care of that for you by bringing the explanations down to earth.

The content is separated into several chapters that contain detailed sections. Each chapter builds on the previous to provide you with a foundation of understanding for the more complex parts of the content at the end of the e-book. Several sample sections step you through the process involved in, for instance, creating a form and attaching it to a PERL script to submit information. These samples will explain the *how* and *why* of Web site development as well as the details of the actual construction.

As you move through this e-book, you'll learn the basics of Web site development, however, realize that the technology is always changing. Throughout the e-book, I'll also provide you with many references and URLs to help you find additional information. These references allow you to take this basic information and expand your knowledge as technology moves forward.

Good luck in your endeavors!

1. Introduction

The Web developers of today must be literate in a variety of rapidly changing languages and technologies. In some cases, it's also useful for the end-user or manager to understand various Web site development concepts. To help you understand the core concepts of Web site construction and operation, this e-book introduces the markup language for Web documents (HTML) and the tools and technologies used to create interactive content on the Web.

The information presented throughout this e-book is provided in a simple, understandable format. The content is detailed in several chapters to provide an increasing core of knowledge. As presented in the following list, each part provides the basic knowledge to understand each subsequent chapter.

Web Site Overview	Discusses the details of how a Web site works including hardware, software, and the interactions that occur between a browser and the Web site.
Basic HTML	Presents the most basic elements of a Web page and what these elements represent.
Advanced HTML	Presents advanced HTML programming topics including graphics, frames, forms, and tables.
JavaScript	For your first step into advanced programming techniques, we'll discuss and build a JavaScript application for your Web pages.
CGI Programming	Discusses CGI programming for your Web pages. We won't develop any new code as that is beyond the scope of this e-book, but you'll learn how to integrate existing code into your Web pages.
ASP Programming	Active Server Pages are an important topic as many Web sites use this technology for interactive content. We'll build a few pages so that you can learn how this technology works.
Java Applets	Java programming is a huge area in the Web site development arena; however, to understand how to develop Java applets would take another massive e-book. This chapter will, however, present the various aspects of using Java applets in your Web page.
Advanced Programming	Once you've grasped the information presented this far, we'll discuss advanced Web development techniques and technologies. The information presented won't make you an expert, but it will allow you to understand how the advanced technologies work with the information presented earlier in the e-book.

Remotely Hosted Applications

Remotely hosted applications are another aspect of Web development that provides tools and services for your site. In this chapter, we'll examine various remotely hosted applications and how you can integrate these applications into your site.

Putting It All Together

We'll reexamine everything discussed in the e-book and provide an overview of how to get your Web site up and running.

Appendix

The appendices provide general information and links in a simple reference. This information is available to help you as you begin developing your Web site.

2. Web Site Overview

A Web site consists of various applications that provide communications with client browsers. Using various technologies, a Web site can provide interactive content using back-end data as well as information provided by the user.

2.1 What is the Web?

The World Wide Web, or simply *The Web*, is a distributed, hypertext-based information system developed at CERN (<http://www.cern.ch/>) that uses various standard technologies. The following list and Figure 1 provide a general overview of the Web and its underlying technologies:

- The underlying network (i.e., Internet, corporate intranet) provides the communications medium over which the Web operates.
- The HyperText Transfer Protocol (HTTP) provides a common application-level protocol for the transfer of data.
- The HyperText Markup Language (HTML) provides a standard presentation-level standard format for describing the structure of documents.
- Servers connected to the network respond to requests from browsers.
- Browsers provide the mechanism for users to access and view hypertext documents.

Browsers also use other standard protocols (i.e., FTP, NNTP, WAIS, gopher) for file transfer and Usenet access.

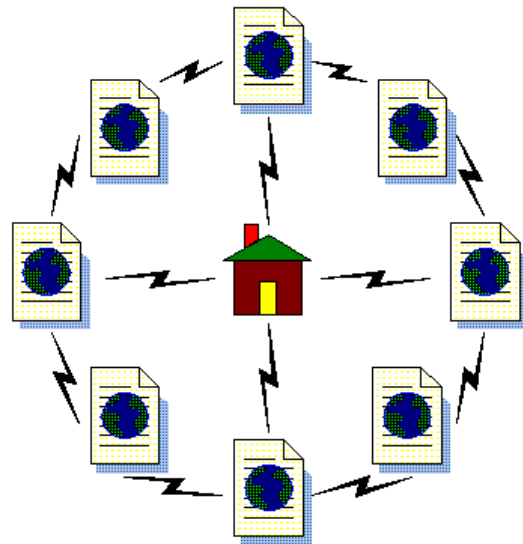


Figure 1: The World Wide Web

2.2 How does it work?

The basic steps involved in Web-based communications are as follows:

- The user enters a URL into a browser or clicks on a link on a page to activate a URL.

- The browser resolves the address of the server noted in the URL using a configuration specific Domain Name Server (DNS).
- The browser connects to the server.
- The browser issues a request for a page to the server.
- The server sends the HTML of the requested page to the browser.
- The browser builds the textual content and requests any additional images or applications (e.g., Java applets).
- The server sends the requested information.
- The browser displays the sent information.

Of course, as with anything, the underlying operation is much more complicated. The intricacies of the operation are, however, hidden so that you only have to do a few things to make a page display in your browser.

From Figure 2, a **Web Client** is any host machine that runs a Web browser or other application that can communicate with a **Web Server**. The Web Server runs software that provides services to any requesting client. The image depicts data flow from the Web Server to the Web Client.

Presentation Layer

The Presentation Layer is the end of the line for Web-based communications. This is the point where the information submitted over the Web is reduced to the tags and basic HTML we see in our browsers.

On the server, this layer is represented by a *data store*. This store contains static HTML pages, dynamic templates, applets, and images. As the client browser requests pages and information, the server software takes the information from this store to return to the browser.

On the client, the Presentation Layer represents the browser itself. At this level, the information is converted to the formatted content, images, and applications that we see on our PC.

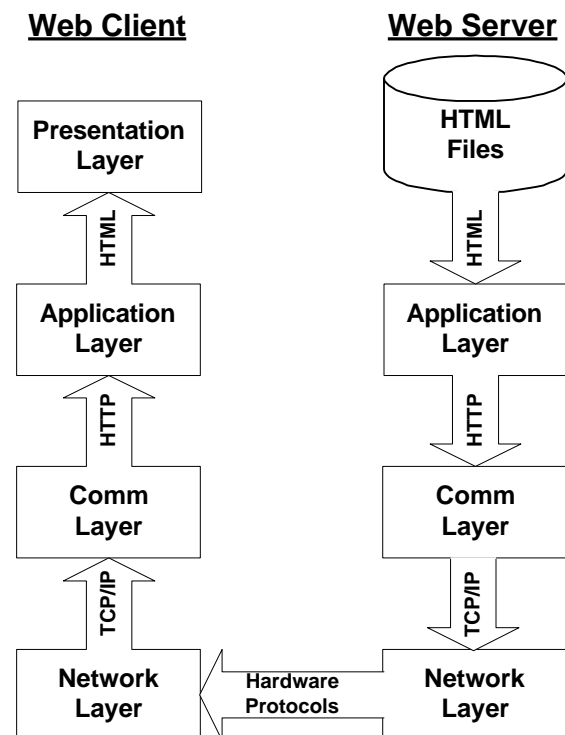


Figure 2: Web Communications

A Web Client initiates a request by submitting a command to the Web Server, for example, for a specific page. This request is initiated when the user of the Web Client enters a URL into the Web browser to request a page from the specified host. When the Web Server responds, it packages HTML into a format that can be sent over the next layer of communications, the HTTP protocol. This packaging allows the server to send the information in a known format to any requesting browser.

Application Layer

The Application Layer on the server contains any software that provides a way of servicing requests from remote clients. This layer usually consists of Web Server software (e.g., Microsoft Internet Information Server, Netscape Enterprise Server, etc.) The server software is responsible for accepting HTTP requests and packaging HTML data into HTTP packets for submission back to the client-side application.

On the client, the layer consists of a browser or any software that can make requests to a Web Server and accept responses. The client-side software is responsible for packaging user requests (e.g., URLs, link clicks) and submitting them as HTTP requests to the server. The software also accepts HTTP packets from the server and extracts HTML, images, and applications for presentation and use in the client-side Presentation Layer.

Comm Layer

The Comm Layer exchanges HTTP packages on the client and server and manages the protocol used for communications between two or more machines. At this level, the protocol consists of TCP/IP, DECNet, or any protocol that allows communications over any network. The primary purpose of this layer is to provide an efficient, error-free communications medium for the higher-level applications.

For example, if the URL `http://www.toupin.com/index.html` is entered into a Web browser, the browser will first locate the host (i.e., `www.toupin.com`) by resolving the address with a Domain Name Server (DNS). The browser resolves the address by sending the host name to the DNS server, provided by the user's ISP. The DNS server will attempt to locate the numeric address of the host and return the address to the Web Client.

If the address is found, the browser connects to the Web Server using an HTTP connection. This connection is a network request by the browser to initiate communications with the server on port 80, 81, or any specified port number. The port is a unique service point that identifies an entry-point on a server that is using a specific protocol. This is similar to having three doorways into a room, with each one leading to a person that translates a discussion in that room using a different language. If an English speaking person enters the Japanese doorway, then of course, the English person would not be able to understand the discussion.

For ports, if a client makes a request to a port number that does not use the same protocol (i.e., speak the same language), there can be no communication. In this case, port 80 is the standard port through which Web-based communications occurs.

As demonstrated in Figure 3, each individual port corresponds to a particular protocol handler. The client applications know the port numbers as part of their design and configuration. For instance, when an e-mail client checks for new messages, it connects to port

110 of a mail server. The POP3 protocol handler processes the requests and sends the mail client the e-mail in the respective user's mailbox. This same type of processing occurs for all protocols used on the Internet.

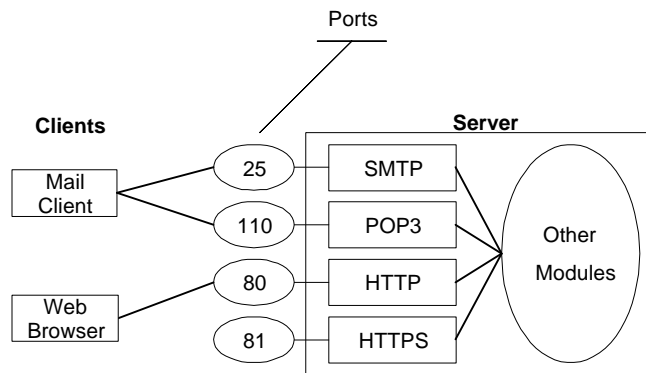


Figure 3: Internet Server Ports

Network Layer

The Network Layer performs the actual packaging that allows the communications protocol to travel over the electrical connection of a network. At this level, the client and server machines merely see a series of pulses on a wire. It's the job of this layer to convert those pulses of electricity into packaged information that can be understood by the Comm Layer. Such packaging can be for Ethernet, Token Ring, X.25, or any low-level network medium that physically connects the machines.

2.3 Web Site Software

The basic architecture required to establish Web-based communications consists of, at least, a Web server and a browser. Adding back-end processors to provide additional features, however, can enhance the functionality of these applications.

The Web Server

The Web Server consists of a piece of software that services requests from a Web Client. This application provides a way to monitor port 80, for most installations, to accept requests from remote clients. The server handles all front-end processing for returning Web pages and information to the requesting client. All back-end processing, such as database access and dynamic pages, is handled by additional applications.

The core Web Server software usually consists of one of the following vendors' applications:

- **Apache:** Usually runs on a Linux or UNIX system and is becoming the more common installation (<http://www.apache.org/>).
- **Microsoft:** Internet Information Server and Site Server are the standards for most organizations running Windows NT servers (<http://www.microsoft.com/>).
- **Netscape:** The Netscape Enterprise Server or iPlanet are present on various UNIX and Windows NT systems (<http://www.netscape.com/>).

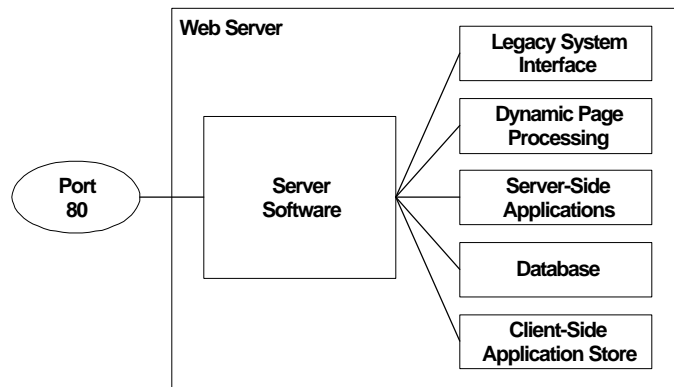


Figure 4: Web Server

The back-end software can consist of one or more of those categories presented in Figure 4. These back-end systems provide a way to enhance the functionality of the core Web Server by adding features that can provide a full application suite for the end user. Such back-end applications include:

- **Cold-Fusion:** Dynamic Web page generation (<http://www.allaire.com/>).
- **MS-SQL, Oracle, Sybase, ODBC, JDBC:** Database access (<http://www.microsoft.com/>, <http://www.oracle.com/>, <http://www.sybase.com/>).
- **Java Applets:** Downloadable client-side applets (<http://www.sun.com/>).
- **PERL, Java, C, C++, Visual Basic:** Customized server-side applications (<http://www.perl.org/>, <http://www.sun.com/>, <http://www.microsoft.com/>).
- **Legacy Applications:** Various interfaces to enterprise systems to provide information for Web-based presentation.

The Web Client

The Web Client is any application that issues requests to a Web Server and can manage the responses and returned information. Once the data is presented to the Web Client, the browser must be able to format and display the HTML and graphics as well as execute Java applets and various embedded scripts within the page. The back-end support introduces subordinate environments that allow applets and scripts to run properly in the browser.

The browser software can be any standard browser, as provided in the following list:

- Netscape Navigator (<http://www.netscape.com/>)

- AOL (<http://www.aol.com/>)
- Internet Explorer (<http://www.microsoft.com/>)

Additionally, the browser from Figure 5 can also be any application that can issue requests and manage communications with a remote Web Server using the HTTP protocol. Such applications provide a means of automating the data retrieval process by downloading pages and parsing the data to use in other applications or store in databases.

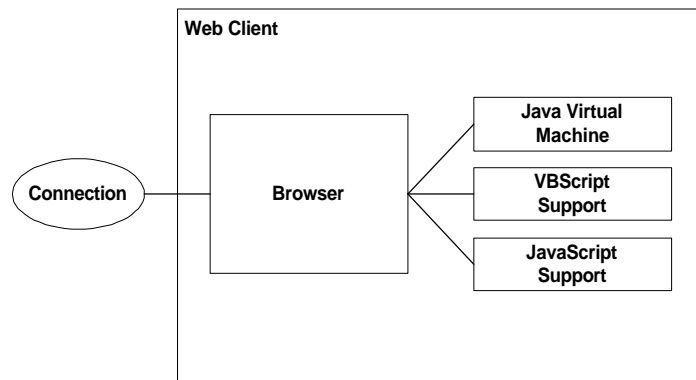


Figure 5: Web Client

2.4 Obtaining a Web Site

Web sites comes in many shapes and sizes: from virtual hosts and dedicated servers to free space. Whatever you choose is important to your presence, however, make sure your choice meets your needs and your desired presence. For more information on domain names and their selection, refer to *Web Site Design and Marketing Strategies*.

Virtual Hosting

Many ISPs provide a virtual hosting service for their customers. In this way, one host can provide unique domain space for each individual customer.

One of the most common ways that an ISP handles virtual hosting is by using the requested host name and the IP address. For instance, when a request for `http://www.host.com` comes into a Web server, the server can determine the target virtual domain based on the host name (i.e., `host`) submitted during the request.

From Figure 6, a remote browser issued a request for `http://www.toupin.com`. The browser located the host, `www.isp.com`, and resolved the host's IP address using the domain name server (DNS). The request is then sent to the server at the designated IP address.

The host, `www.isp.com`, contains four virtual domains, each of which contains its own directory and content. Once the server accepts the request, it uses the host name, `toupin`, to determine which virtual domain directory to access. The server then loads the requested information from the specified directory and submits it back to the requesting browser.

Using this virtual domain approach to hosting, you can setup a Web site on a single host with a single IP address and share that host with thousands of other users.

Those that surf to the site are unaware, for the most part, that your site is located on an ISP as the virtual hosting operations are transparent to the surfer. This approach is the most efficient and professional for the price. In most cases, the cost of such hosting ranges between \$19.00 and \$75.00 per month.

Before selecting an ISP, here are a few pointers:

- **Determine Storage Space Needs**

The amount of room you need depends on what kind of site you'll setup---the average Web site uses about five megabytes of space or less.

- **Determine Required Bandwidth**

Bandwidth is the amount of data transferred to or from a Web site. Some Web hosting companies often limit the amount of bandwidth accessible for each site to appropriately share bandwidth with other users of the company's resources.

- **Determine the Necessary Speed of Your Site**

The connection that your Web hosting company has to the Internet will determine how quickly and easily your site's content can be viewed by visitors. Web hosting companies with faster connections don't always cost more than those with slower connections.

- **Research the Quality of Technical Support**

Examine all aspects of the support services available and test them to make sure they can deliver appropriate solutions when you have a problem.

- **Research the E-mail Services**

Commonly available e-mail services for you to consider when selecting a Web

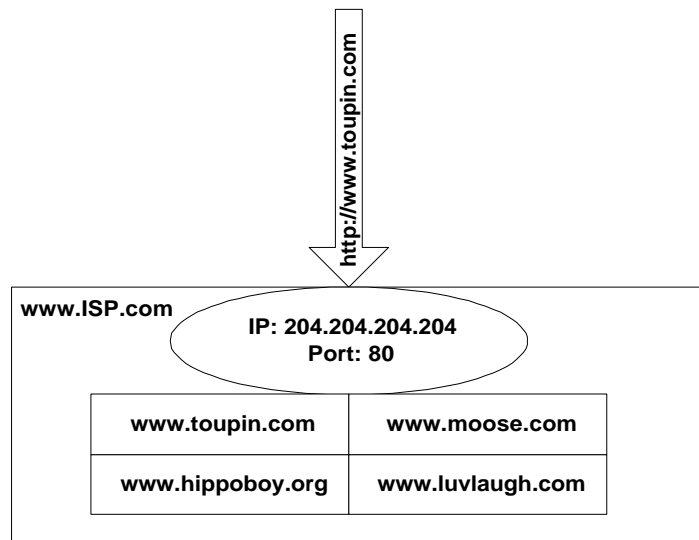


Figure 6: Virtual Hosting

host include POP3 (post office protocol) accounts, forwarding, autoresponders, mailing list support, and Web accessible e-mail.

- **Evaluate Web Site Monitoring Tools**

Web hosts provide many options to help you maintain your site, such as control panels, text files, and e-mail or phone requests.

- **Research Web Site Accessibility and Modification Options**

To access and modify a Web site, you may need to transfer files, make changes from a remote terminal, or find an easy and simple way to alter your Web site.

These functions can be accomplished with the file transfer protocol (FTP), Telnet, and Microsoft FrontPage applications.

- **Research Extra Services**

Some hosting companies use additional services to make a larger impact on the market than their competition. Some services range from domain name parking to non-profit discounts and contract programming.

- **Determine Fee Structure**

Web hosting companies differ in costs, services, setup fees, and guarantees.

Some site hosting is free, while other hosts charge per 50 MBs. Setup fees, registration fees, annual rates, and guarantees all depend on which host you choose. Determining the financial details of different Web hosts, especially the one you decide to use, is necessary to avoid surprises later on.

For a list of available ISPs nationwide, check out ISP.com (<http://www.isp.com/>).

Dedicated Server

A dedicated server is much more expensive to maintain, however, it'll provide you with your own domain space. One major problem, especially if you're an individual or small company trying to setup a server, is that of ensuring uptime for the server. Two approaches for setting up your own server are *co-location* and *onsite hosting*.

Co-location

Co-location involves purchasing a server, setting it up, and letting another organization monitor and maintain security for the server. The hosting center provides the connection to your network or it can provide you with a network connection to your local network or to the Internet. You can contact one of the many co-location services on the Internet to find out pricing and necessary setup information.

Onsite Hosting

Onsite hosting allows you to have your server in your office or home and involves some expense and setup.

First, setup your server with a Web server software and a mail server software---this allows your server to handle Web server requests. You'll also need an SMTP mail server---and a backup server called a *Secondary MX* or *Secondary Mail eXchanger*. Sendmail on Linux is popular. NT also has an SMTP mail server, and Qualcomm's Eudora has one for Macs and PCs. You can usually make your secondary Domain Name Server (DNS) the backup mail server, but make sure there's enough disk space to handle the mail that'll land on the secondary when the primary mail server is down for maintenance.

Next, you'll have to acquire one or more IP addresses for your servers. This IP address is the numeric address of the host and is associated with the name of the host. You also need a moderate-speed, full-time Internet connection with static IP addressing---that is, an address or range of addresses that does not change. All of this can be obtained from an ISP. You don't need a full T-1 as a fractional T-1, ISDN, or DSL connection will work.

You'll also have to setup the domain to resolve IP addresses by setting up the *Domain Name Server*. This server allows you to add your host name and the host's IP address to the array of DNS's on the Internet. Another option is to have another DNS, such as an ISP, host your records. If you can't find an ISP to host your records, refer to the Public DNS (<http://www.granitecanyon.com/>). You'll also need someone to provide *Secondary DNS*, or you can do that yourself with a second computer---this is the backup nameserver when your main system is down.

Sub-Domains

Sub-domains usually indicate that you're getting your hosting service free or for a lesser cost than most domains. This approach allows you to pay a fraction of the cost of a standard domain; however, the hosting company gets free advertising.

A sub-domain is similar to virtual hosting except that the domain name also contains the name of the hosting company. For instance, the subdomain URL would appear as `http://user.host.com`.

Some sub-domain hosting companies are:

- 1Avenue (<http://www.1avenue.com/index2.html>)
- 00Homepage (<http://www.00homepage.com/>)
- TopCities (<http://www.topcities.com/>)

Free Space

Of course, there's always free space. The Web host makes money by either placing ads on your site, or taking a percentage of the actual call costs from the telephone company. This is fine for a personal home page or a local club or society page, but it does cause problems if you're trying to present a professional image.

Some free hosting companies are:

- DenCity (<http://www.dencity.com/>)
- Yahoo! Geocities (<http://geocities.yahoo.com/home/>)
- Northern Skies (<http://communities.northsky.com/>)
- 008Web.com (<http://www.008web.com/>)
- Tripod (<http://www.tripod.lycos.com/>)
- FortuneCity (<http://www.fortunecity.com/>).

Thank you for reading the sample of this e-book!

The remainder of this e-book can be purchased at:

<http://www.onlinebooksinc.com/>

This book was distributed courtesy of:



For your own Unlimited Reading and FREE eBooks today, visit:

<http://www.Free-eBooks.net>

Share this eBook with anyone and everyone automatically by selecting any of options below:



To show your appreciation to the author and help others have wonderful reading experiences and find helpful information too, we'd be very grateful if you'd kindly [post your comments for this book here](#).



COPYRIGHT INFORMATION

Free-eBooks.net respects the intellectual property of others. When a book's copyright owner submits their work to Free-eBooks.net, they are granting us permission to distribute such material. Unless otherwise stated in this book, this permission is not passed onto others. As such, redistributing this book without the copyright owner's permission can constitute copyright infringement. If you believe that your work has been used in a manner that constitutes copyright infringement, please follow our Notice and Procedure for Making Claims of Copyright Infringement as seen in our Terms of Service here:

<http://www.free-ebooks.net/tos.html>



STOP DREAMING AND BECOME AN AUTHOR YOURSELF TODAY!

It's Free, Easy and Fun!

At our sister website, Foboko.com, we provide you with a free 'Social Publishing Wizard' which guides you every step of the eBook creation/writing process and let's your friends or the entire community help along the way!

LOGON ONTO FOBOKO.COM

↪ and get your story told!

FOBOKO.COM