

---

**AlpineWeb.com**

**Virtual Private Server v1 to  
Virtual Private Server v2  
Self Migration Process  
for Virtual Private Server v2  
Administrators**

**First Edition  
September, 2003**

---

# Table of Contents

<b>Introduction .....</b>	<b>1</b>
Starting the Shadow Process.....	1
<b>Creating Users .....</b>	<b>2</b>
<b>Adding Virtual Hosts and Other Apache Changes .....</b>	<b>2</b>
<b>Migrating Sendmail and Mail programs .....</b>	<b>4</b>
<b>Vinstalling Additional Add-ons .....</b>	<b>6</b>
Perl Modules .....	6
<b>Migrating Databases .....</b>	<b>7</b>
<b>Moving Your Custom Digital Certificate .....</b>	<b>8</b>
<b>Migrating Time-Sensitive Material .....</b>	<b>9</b>
<b>Document Conventions .....</b>	<b>11</b>

## Introduction

The shadow process for a Virtual Private Server v1 to a Virtual Private Server v2 server is meant to be a self-migration for users with UNIX experience. This document does not explain the differences between a Virtual Private Server v1 and a Virtual Private Server v2 server. However, this document steps you through the basics of how to migrate your Virtual Private Server v1 server to a Virtual Private Server v2 server.

If you have made significant changes to your Virtual Private Server v1 server, this document may not be helpful.

The Virtual Private Server v1 to Virtual Private Server v2 migration process takes several hours, maybe even days, depending on your knowledge of the Virtual Private Server v1 and Virtual Private Server v2 servers, and the amount of changes you have made to a standard Virtual Private Server v1 server.

When you have successfully shadowed your account, you will receive a confirmation email with your shadow account information, including your shadowed account's IP address.

When connecting to your Virtual Private Server v2 account to perform migration tasks, you will need to connect as the root user using the IP address of the shadowed account.

## Starting the Shadow Process

The Virtual Private Server v1 ("old") to VPS2 ("new") migration procedure begins in the VPS2 Migration Script.

1. Open a browser and go to the Hosting Plan Upgrade Order form:  
<https://secure.alpineweb.com/order/hosting/upgrade.html>
2. Fill in your account details, the plan you're upgrading from, the VPS2 plan you are upgrading to and then press the Submit button.

The Shadow process creates a new VPS2 account with same account owner (the administrative user) and a root user. The root and account owner passwords will be the same as the Virtual Private Server v1 account owners. The following files and directories are created on your new VPS2 server:

- `/.migrate` – directory
- `/.migrate/bin` – directory
- `/.migrate/accountname` – directory containing copy of file system from old account
- `/.migrate/bin/addaccounts.pl` – file for creating user accounts
- `/.migrate/bin/migratesm` – file for migrating Sendmail
- `/.migrate/bin/README` – instructions for `addaccounts.pl` and `migratesm`

---

## Creating Users

Your first task after the shadow process is to create your users by running the `addaccounts.pl` script. To do this:

1. SSH to your new VPS2 server as root, using the IP address of the shadowed account. In the command line, type:

```
% cd /.migrate/bin
% ./addaccounts.pl accountname
```

where “accountname” is your old Virtual Private Server v1 account owner.

This script creates all the user accounts that were on the old server and their home directories at `/home/username` on the new server, with FTP and mail access and no shell access. Users with FTP and mail access will be added to the `ftp` and `mail` groups respectively, and each user will have a group created for them. The quota for each user remains as it was. If you want to grant shell access to a user, run the `pw` command. An example follows:

```
% pw usermod joe -s /bin/csh
```

2. To check the amount of disk space being used by a specific user on your server, type:

```
% quota username
```

where “username” is the name of your user

## Adding Virtual Hosts and Other Apache Changes

After you have added the users, re-create the Virtual Hosts that you are hosting. In the past Virtual Hosts were generally stored in the `vhosts` directory for Virtual Private Server v1. For VPS2 we *strongly* suggest that you associate each Virtual Host with a user and put the Virtual Host site under that user’s home directory. For example, if you have a Virtual Host for `test2.com` and bob owns that site you would want to add the site to user bob’s home directory.

You cannot simply copy over the `httpd.conf` file or the VirtualHost blocks of the `httpd.conf` file. The format for each is completely different between a Virtual Private Server v1 and a VPS2 server. You will need to run `vaddhost` again for each Virtual Host.

An example follows of how to add a Virtual Host with the best settings.

1. Connect to your VPS2 server using SSH and type `vaddhost`.
2. Proceed through the script, supplying the requested 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`.
  - 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.

3. Make any additional changes to the `httpd.conf` file that you may have made on your Virtual Private Server v1 account. Do *not* just copy the `httpd.conf` file.
4. Copy the contents of each hosts `htdocs` directory and `cgi-bin` directory.
5. To copy the `htdocs` directory, go to the `./migrate/accountname/www/vhosts/www.test1.com/htdocs/`. Type the following to copy the contents:  

```
% cd ./migrate/accountname/www/vhosts/www.test1.com/htdocs/  
% cp -rp * /home/user/www/test1.com/
```
6. To copy the `cgi-bin` directory, go to the `./migrate/accountname/www/vhosts/www.test1.com/cgi-bin/`. Type the following to copy the contents:  

```
% cd ./migrate/accountname/www/vhosts/www.test1.com/cgi-bin/  
% cp -rp * /home/user/www/cgi-bin/
```

The log files can be copied over to the user's home directory if desired.
7. If you are using FrontPage Extensions, install these now by typing:  

```
% cd  
% vinstall frontpage
```

Be sure to install the extensions for each Virtual Host who uses FrontPage.
8. If you are using PHP you will need to it now by typing:  

```
% cd  
% vinstall php4
```

Be sure to follow the instructions thoroughly and choosing the needed components to install.
9. If you are using any other programs such as `mod_rewrite` or other apache modules, you will need to install these at this time as well.

---

# Migrating Sendmail and Mail programs

If you are using Procmail, Spamassassin, Majordomo, or any other mail program that needs to make modifications to the sendmail.cf file, you need to:

1. Do a vinstall of these programs first. To see a list of available vinstalls type the following at the shell command line:  

```
% vinstall -l
```
2. If you have made any changes to the configuration files for any of these programs, make the same changes now.
3. Additionally, if you have made any changes to your .procmailrc files, copy those changes to the new .procmailrc file as well.
4. Run the migratesm script located at /.migrate/bin/migratesm. This will create your aliases, virtmaps, access, and local-host-names files as well as restart Sendmail to make the changes take effect.
5. If you have any Majordomo mailing list in the default location of ~/usr/local/majordomo/Lists/, those lists will be copied over to the VPS2 default location of /usr/local/majordomo/lists/.
6. When the migratesm script has finished running, an ALIASES\_TODO file is created with a list of mailing lists, programs, and other files that appear on the right-hand-side of the new aliases file. Examine that file to verify that those files are in the same location on your VPS2 as they were on Virtual Private Server v1.
7. When you have completed this, run the migratesm script a second time using the following syntax:

```
% migratesm -intodo
```

This will interpret the input ALIASWS\_TODO lines as follows

- Lines beginning with a pound sign (#) are ignored
- Lines of the form "alias: Y RHS" (where the "center" field is a literal "Y") are ignored. (The "Y" means that the file named in the RHS already exists on the VPS2 server, and doesn't need to be copied from the Virtual Private Server v1 location mentioned in the RHS.)
- Lines of the form "alias: N RHS" (where the "center" field is a literal "N") cause migratesm to attempt to copy the file mentioned in RHS to the same location in the VPS2 file system.
- If the copy succeeds, migratesm changes the "N" in the middle field to "Y" in the updated ALIASES\_TODO.
- If the copy fails because the file does not exist in the Virtual Private Server v1 directory structure, migratesm changes the "N" in the middle field to "N\*".
- Lines of the form "alias: N\* RHS" (where the "center" field is a literal "N" followed by an asterisk [\*]) are ignored.
- Lines of the form "--DESTDIR: /dir/name" (where the line begins with the literal string "--DESTDIR:" and is followed by the path to a VPS2 directory) override where files listed in subsequent line(s) are copied to. (The files listed in subsequent lines are copied from the location named in the third field of the line, to the directory listed following --DESTDIR.)
- Lines of the form "--DFLTDEST" cancel the effect of preceding --DESTDIR lines, causing the source and destination directories of subsequent lines to be the same.
- All other lines are ignored.

The migratesm script updates ALIASES\_TODO and backs up the previous copy in /etc/mail/ALIASES\_TODO.0. Additionally, if you have any Majordomo mailing list in the default location of ~/usr/local/majordomo/Lists/, those lists will be copied to the VPS2 default location of /usr/local/majordomo/lists/.

8. When you have finished, check the following files for errors:

- /etc/mail/aliases
- /etc/mail/virtusertable
- /etc/mail/access
- /etc/mail/local-host-names
- If you are using Majordomo /usr/local/majordomo/lists/

For additional information on how to use the migratesm script, type the following at the shell command line as a non-root user:

```
% perldoc /.migrate/bin/migratesm
```

## Vinstalling Additional Add-ons

Since your server is a new server, you will need to vinstall any additional programs that you may have installed on your Virtual Private Server v1 account. There may not be a vinstall for some of the programs that you had installed in the Virtual Private Server v1 account with vinstalls. If this is the case, you can install the program using the FreeBSD Ports collection. For information on how to install programs using the Ports collection, see the Reseller Handbook, page 190.

## Perl Modules

If you are using Perl and have installed modules that you use, you must install these modules on your VPS2 server as well. VPS2 no longer uses vcpan as the default method of installing Perl modules. Instead, VPS2 now uses the default Perl method of installing modules. To install a module using CPAN:

1. Type the following at the shell prompt:

```
% perl -MCPAN -e shell
```

The first time you run CPAN, you will be prompted for your preferences regarding where to download your modules and other settings. For most settings, accept the default.

2. When it comes time to select your mirror preferences, just select 4 or 5 mirrors that look good to you (you don't have to know anything about them or where they are). If you are really worried about selecting the right mirrors, choose the first 5 (1 2 3 4 5).

Once you have configured your CPAN settings, you'll likely never have to do it again.

3. Once at the CPAN command line (cpan> ), to install a module type:

```
% cpan> install <Module_name>
```

This will get and install the module. To save space on your server, type the following at the CPAN command line:

```
% cpan> clean <Module_name>
```

This will delete all unnecessary files that were used during installation.

---

# Migrating Databases

Only MySQL, the most popular database that is used with your servers, is covered.

1. Vinstall the MySQL database by typing the following at the shell prompt:  

```
% vinstall mysql
```

The vinstall will prompt you whether to install MySQL using the Ports collection or Virtual Private Server v1 style. For convenience, choose the Virtual Private Server v1 style of installation. This will install MySQL and start the MySQL daemon for you.
  2. After MySQL is installed, change the password for the root user by typing the following at the shell prompt:  

```
% mysqladmin -u root password <your-new-password-here>
```

In addition, if you have other users you will need to create the extra users at this point with their proper permissions.
  3. Create the databases from the Virtual Private Server v1 MySQL database. To do this, type the following at the shell prompt for each database:  

```
% mysqladmin create <database-name-here>
```
  4. After creating the databases, migrate your databases from MySQL. To do this, SSH into your Virtual Private Server v1 server using the IP address.
  5. Perform a mysqldump on each database by typing the following at the shell command line:  

```
% mysqldump -u username -p databasename > filename
```

where “username” and “filename” are replaced with appropriate names.
  6. Type the password.
  7. The name of the file should appear with all the SQL and data needed to recreate the database structure and data.
  8. Copy the dump file to the VPS2 server by typing the following:  

```
% scp dumpfilename root@domainname:/root/
```

This will copy the file to the /root directory of the VPS2 server. Do this for each database that you have.
  9. SSH into your VPS2 server and type the following to create the database structure and enter the data into the database:  

```
% mysql -u username -p databasename < /root/filename
```
  10. Type the password.
  11. Do this for each database that you have. After you have finished this, verify that the information was imported correctly.
  12. Because databases contain time sensitive material, delete the databases you just created by typing the following at the command line:  

```
% mysqladmin -u [user] -p drop [database_name]
```

where username is the username of the database and databasename is the name of the database you are deleting.
1. After you have deleted the databases, you will want to recreate them again following step 3 above. However, you will not want to create the dump file or import the information from the MySQL database from the Virtual Private Server v1 server until you are ready to make your VPS2 server live. For more information on perform a MySQL dump and import go to: <http://www.mysql.com/doc/en/index.html>

---

# Moving Your Custom Digital Certificate

The default digital certificate is already on your new Virtual Private Server v1 server. This procedure applies only to custom certificate.

1. Using FTP or another method, copy the certificate and Private Key files to the new server. Both the certificate and the key are stored in the `~/etc/` directory of the Virtual Private Server v1 server. The certificate should be in a file named `ssl.cert`, and the key should be in the `ssl.pk` file. If you use FTP, be sure to copy the files to the new server as ASCII files. You will need to copy them to the `/usr/local/apache/conf/` directory on the VPS2 server.
2. It is a good idea to check your Private Key to make sure it has been decrypted. Use more or your favorite text editor to view the file. If your key has been decrypted, you should not see the following lines before the encoded elements of the key.

```
Proc-Type: 4, ENCRYPTED
DEK-Info: DES-EDE3-CBC, BCC23A5E16582F3D
```

3. If your Private Key does have those lines near the beginning, run the following command to remove the encryption.  

```
% openssl rsa -in ssl.pk -out ssl.pk
```
4. Edit your `httpd.conf` file (located in the `/usr/local/apache/conf/` directory) to look for your certificate files. The following directives need to be added to the main section of your `httpd.conf` file.  

```
SSLCertificateFile /usr/local/apache/conf/ssl.cert
SSLCertificateKeyFile /usr/local/apache/conf/ssl.pk
```
5. After you have added the certificate directives to your `httpd.conf` file, you need to run `restart_apache` to make Apache start using the new certificate.
6. Ensure that 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
```
7. 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.

# Migrating Time-Sensitive Material

There are three areas requiring migration that are time sensitive:

- DNS
- Mail
- Databases

These have to be migrated with minimum downtime and without losing information. For this reason, they migrated last. Perform the following steps precisely in order to achieve the best results.

1. SSH into your Virtual Private Server v1 server and add the following line to the `~/etc/hosts.allow` file using your favorite text editor.
 

```
SMTP submission aol : all : twist /bin/echo "450 This server
is moving its IP address!"
```

This will block all mail to your Virtual Private Server v1 server. All mail will be bounced with a 450 error that will tell the sending mail server to retry sending later. This way, you should not lose any mail.
2. In the VPS2 Migration Script, change your main Address Record (A) for each domain on your server to the new IP address of the VPS2 server. This will also change the Time to Live (TTL) for each domain up to 24 hours from 10 seconds. If there is a problem and you need to move your domain back to the old server, you will need to contact support to do this. In addition, you will want to have support change your TTL back to 10 seconds. If any DNS servers have cached their DNS before you have support change your DNS back and your TTL, it will take those DNS servers 24 hours to update you DNS information again.

To avoid all this you can have support move your domains over to the new VPS2 server and make sure they keep the TTL to 10 seconds. If there is a problem, then you can call them back and have them reverse it keeping the TTL at 10 seconds.

3. While still connected to your VPS2 server rename each user's mail folder, for example:
 

```
% cd /var/mail
% mv username username_new
```

Do this for every user except root, since there is no root user on the Virtual Private Server v1 server.
4. Copy the inbox or `/var/mail/username` for each user by typing the following:
 

```
% cd /var/mail
% scp accountowner@<IPofVirtual Private Server
vlserver>:"var/mail/*" .
```

This will copy all the files in the Virtual Private Server v1 server's `~/var/mail` directory, which should be a list of the accounts on the Virtual Private Server v1 server.
5. Next, you will need to append any mail on the new VPS2 server to mail from the Virtual Private Server v1 server by typing the following for each user on the server, except root.
 

```
% cd /var/mail
% cat username_new >> username
% rm username_new
```
6. Using your favorite file transfer tool, copy any IMAP folders from the user's home directory on the Virtual Private Server v1 server to the user's home directory on the

VPS2 server. You must now use the IP address of the Virtual Private Server v1 server instead of the domain, since the domain will now point to the VPS2 server instead.

You will need to do this manually since there may be many different names for mail folders.

7. Migrate your databases from MySQL. To do this, SSH into your Virtual Private Server v1 server using the IP address.

8. Perform a mysqldump on each database by typing the following at the shell command line:

```
% mysqldump -u username -p databasename > filename
```

where “username” and “filename” are replaced with appropriate names.

9. Type the password.
10. The name of the file should appear with all the SQL and data needed to recreate the database structure and data.

11. Copy the dump file to the VPS2 server by typing the following:

```
% scp dumpfilename root@domainname:/root/
```

This will copy the file to the /root directory of the VPS2 server. Do this for each database that you have.

12. SSH into your VPS2 server and type the following to create the database structure and enter the data into the database:

```
% mysql -u username -p databasename < /root/filename
```

13. Type the password.
14. Do this for each database that you have. After you have finished this, verify that the information was imported correctly.
15. After you have completed all the steps above, verify that all services are functioning properly. In addition, check with all your users to verify they have all the files they need. Also, check all Web pages to ensure they are functioning properly.

# Document Conventions

The conventions used in this document are designed to be completely predictable and are used for the following specific purposes.

## Conventions List

### Typeface

*Italic*

### Usage

Used to indicate the following:

- The first mention of new terms in any information unit. For example:  
The *rudaplex* and the *strataguide* have been the modified for this model.
- References to titles of books, chapters, headings, CDs, diskettes, or software programs. For example:  
Refer to *The Technical Manual* for technical term descriptions.
- Variables that the user types. For example:  
Type the *User ID* in the User ID text box.

### Bold

Used to indicate the following:

- Exact text strings typed. For example: Type **ABCDEFGH**.
- Keyboard keys pressed. For example: **Press Ctrl-A**, then press **Enter**.

### Blue Underline

Used to indicate linked email, IP, Network, or Web addresses. For example:  
Go to <http://www.microsoft.com> for more information about Microsoft products.

### Cross-Reference

Used to indicate a reference to another part of the same document. The grey portion of the cross-reference is hot linked to the appropriate section of the document, followed by a page number, also hot-linked to the same portion of the document. For example:  
For more information about the Document Conventions, see the "Document Conventions".

### Operating System Text

Used to indicate text that appears in a shell session for an operating system. The displayed text pertains to operating system text only, not application elements. For example:  
Type LIST MAIN FOLDER. The screen displays the Main folder.

### Program Code

Used to indicate code listings. For example:

```
{
# do something;
}
# check to see if $user has the attrib 'attrib'
if (hasKey($user_obj, 'attrib', $dbh) != 1)
{
print "User not Authorized to update!";25
}
```

### Screen Element

Screen elements consist of anything that is displayed on screen (exclusive of the operating system). This includes toolbar menu items, drop-down lists and items in a drop-down list, buttons, or anything else a user sees on screen. For example:

- From the Printer drop-down list, choose Local Printer. The Are You Sure? dialog box appears. Click OK.
- The following message appears: User Not Authorized

# Special Elements

These elements provide a variety of information ranging from warnings that the reader should not neglect to supplementary information and tips that will simply enhance their reading experience.

---

**Tip** Used to point out helpful ideas, some not-so-obvious features, quick or alternate ways to get a particular job done, and techniques you might not discover by yourself. The **Tip List** special element is used when multiple tips are used.

---

**Note:** Used to highlight certain information for the reader. Generally, the Note element provides additional information on the current topic. The **Notes:** special element is used when multiple notes are required.

**Important:**

Used for information that is considered more pertinent to the reader than information presented in Note elements.

---

**Caution:**

*Used as a hazard light in documents. Information included in a Caution element could save the reader from hours of lost work.*

---