

# Red Hat 9 / Apache 2.x + SSL / PHP 4.3.x / MySQL 4.1.x / osCPro HOWTO

This page is created by "MM Linux User Group" 2005 Feb

## Revision History

Revision 1.5                      2004-06-02                      Revised by: Theo

- Refined instructions for users who are new to Linux
- Fixed an error in MySQL setup (missing daemon startup)
- Fixed an error in PHP parsing. (missing MIME type in apache configuration)
- Added in additional instructions for SSL setup

Revision 1.2                      2004-05-24                      Revised by: Theo

- Removed use of osCommerce
- Modified MySQL install procedures for compability with osCPro.
- Rearranged Table of Contents
- Modified osCPro install procedures to correct a bug that was associated with MySQL.
- Added in commands to have apache start up automatically with SSL.

Revision 1.1                      2004-05-24                      Revised by: Theo

Fixed package version information.

Revision 1.0                      2004-05-17                      Revised by: Theo

First version.

## Table of Contents

- [1. Introduction](#)
  - [2. Required packages](#)
  - [3. Installing Red Hat Linux 9](#)
  - [4. Installing MySQL 4.1.x](#)
  - [5. Installing Apache 2.x](#)
  - [6. Installing PHP 4.3.x](#)
  - [7. Installing Zend Optimizer](#)
  - [8. Install and Create an SSL Certificate](#)
  - [9. Installing osCPro](#)
- 

## Introduction

This document will guide the user to install the Linux operating system (Red Hat 9), Apache 2.x webserver with Secure Socket Layers, MySQL 4.1.x database, PHP 4.3.x, and osCPro shopping cart.

## Required packages

The following packages are required for this tutorial. Unless specified, download the tar'd source for each package. Try to avoid RPM/SRPM.

## Red Hat Linux 9

All CDs required.

<http://www.redhat.com>

## Apache 2.0.49

<http://httpd.apache.org/>

## mySQL 4.1.1

<http://dev.mysql.com/downloads/mysql/4.1.html#Source>

## PHP 4.3.x

<http://www.php.net/downloads.php>

## Latest Version of CGI.pm and DBI

<http://www.cpan.org/modules/by-module/CGI/>

<http://www.cpan.org/modules/by-module/DBI/>

## osCPro

<http://www.oscpro.com/shop/index.php>

## Zend Optimizer

[http://www.zend.com/store/free\\_download.php](http://www.zend.com/store/free_download.php)

Make sure all packages will be in the `root` directory of the server.

## Installing Red Hat 9

Install RH9 with a custom install; the development tools must be selected. Do not install the `httpd` and `mysql` packages. Make sure the `curl` package is installed.

## Installing mySQL

### Documentation

<http://dev.mysql.com/doc/mysql/en/index.html>

Make sure you are logged in as `root`. Extract the mySQL package (`tar xfvz filename`) and change into the mySQL source directory.

1. Add the user `mysql` into the group `mysql`

```
> groupadd mysql
> useradd -g mysql mysql
```

2. Change into the mySQL directory and configure mySQL with the following and install

```
> ./configure --with-openssl --with-isam --prefix=/usr/local/mysql
> make
> make install
```

3. Copy the default configuration files to your mySQL directory

```
> cp support-files/my-medium.cnf /etc/my.cnf
```

4. Install the default user database

```
> cd /usr/local/mysql/bin
> ./mysql_install_db --user=mysql
> cd ..
> chown -R root .
> chown -R mysql var
> chgrp -R mysql .
```

5. Start the MySQL server.

```
> cd /usr/local/mysql/bin
> ./mysqld_safe --user=mysql &
```

6. Run the MySQL command line client and secure MySQL.

```
> ./mysql
```

You are now in the MySQL CLI prompt. All SQL commands end with a semicolon.

```
mysql> drop database test;
mysql> use mysql;
mysql> delete from db;
mysql> delete from user where not (host="localhost" and user="root");
mysql> flush privileges;
```

Note: Make sure you spell privileges correctly. If you get any errors when you have to type a line with that word, it is most likely you misspelled it.

7. Add an osCommerce username so we do not have to use a root account for MySQL access. You will be creating two logins, one that is a login for the server itself, and another for remote access.

```
mysql> GRANT ALL PRIVILEGES ON osCommerce.* TO 'osCommerce'@'localhost'
IDENTIFIED BY '';
mysql> GRANT ALL PRIVILEGES ON osCommerce.* TO 'osCommerce'@'%' IDENTIFIED BY '';
```

8. Set the password for osCommerce. Replace your\_password with the password you will be using.

```
mysql> SET PASSWORD FOR 'osCommerce'@'localhost' = OLD_PASSWORD('your_password');
mysql> SET PASSWORD FOR 'osCommerce'@'%' = OLD_PASSWORD('your_password');
mysql> FLUSH PRIVILEGES;
```

9. Set the root account password for MySQL.

```
mysql> \q
```

```
> ./mysqladmin -u root password new-password
```

10. Add MySQL to startup.

```
> cp /usr/local/mysql/share/mysql/mysql.server /etc/init.d/mysql
> chmod +x /etc/init.d/mysql
> chkconfig --add mysql
```

11. Extract the CGI.pm package and install.

```
(extract using tar xfvz filename and change into the CGI.pm source directory)
> perl Makefile.PL
> make
> make install
```

12. Extract the DBI package and install.

```
(extract using tar xfvz filename and change into the DBI source directory and
extra
> perl Makefile.PL
> make
> make install
```

The CGI.pm and DBI packages allow you to run the command line MySQL utilities.

## Installing Apache 2.x

Make sure you are logged in as `root`. Extract the Apache 2 package (`tar xfvz filename`) and change into the Apache 2 source directory.

1. Set the include and library path for Kerberos.

```
> export CFLAGS="-I/usr/kerberos/include/ -L/usr/kerberos/lib"
```

2. Configure Apache 2 with the following and install.

```
> ./configure --enable-so --enable-ssl
> make
> make install
```

- Edit the Apache config files to handle PHP.

```
> vi /usr/local/apache2/conf/httpd.conf
```

The vi editor will now start.

- Edit the `ServerAdmin` line to reflect the administrator's e-mail address.
- Edit `DocumentRoot` to reflect the path where HTML and PHP files will be served.
  - If you changed `DocumentRoot` from the default, you will have to find the line that reads `<Directory "/usr/local/apache2/htdocs">` to the path you used in `DocumentRoot`
- Find the line that says `DirectoryIndex index.html index.html.var` and change it to `DirectoryIndex index.php index.html index.html.var`

Save the file by typing `<esc>`, colon, `wq`, `<enter>`

```
:wq
```

Press `<enter>`.

3. Edit `mime.types` to include handlers for PHP.

```
> vi /usr/local/apache2/conf/mime.types
```

At the bottom of the file, insert the following line:

```
application/x-httpd-php .php
```

4. Modify the script for apache to start up on boot with SSL.

```
> cp /usr/local/apache2/bin/apachectl /etc/rc.d/rc3.d/S58httpd
```

Where `rc3.d` is the current boot run level. If you use a graphical login, change `rc3` or `rc5`

```
> vi /etc/rc.d/rc3.d/S58httpd
```

In vi, find the line that begins with

```
HTTPD='/usr/local/apache2/bin/httpd'
```

Change it to

```
HTTPD='/usr/local/apache2/bin/httpd -DSSL'
```

Save and exit vi.

At this point, do not reboot the machine. Apache will not start since SSL is not completely configured yet. After you complete the SSL setup, when you boot the server, you may be asked for the SSL password on boot; the system will not fill this value in automatically (although there are scripts that will).

## Installing PHP

### Documentation

<http://www.php.net/docs.php>

Make sure you are logged in as `root`. Extract the PHP package (`tar xfvz filename`) and change into the PHP source directory.

1. Configure and install PHP with the following parameters.

```
> ./configure --with-apxs2=/usr/local/apache2/bin/apxs --with-mysql --with-curl
> make
> make install
```

2. Copy the default PHP configuration files

```
> cp php.ini-dist /usr/local/lib/php.ini
```

3. Edit `/usr/local/lib/php.ini` and set `register_globals` to on

```
> vi /usr/local/lib/php.ini
```

The vi editor will now start and open `php.ini`. Scroll down the file until you find `register_globals = Off`. Change "Off" to "On" by pressing insert to go into edit mode. When you are done editing, hit `esc`, and then press the following in order, colon, `wq`, `<enter>`. Your prompt should look like

```
:wq
```

Then press `<enter>` to save your file and quit the editor.

4. Add the appropriate types to the Apache server.

```
> vi /usr/local/apache2/conf/httpd.conf
```

Insert the following lines at the bottom of the file:

```
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
```

Save and quit vi.

## Installing Zend Optimizer

Make sure you are logged in as `root`. Extract the Zend Optimizer package (`tar xfvz filename`) and change into the `Zend Optimizersource` directory.

1. Install Zend Optimizer.

```
> ./install
```

2. An interface should appear for setting up Zend Optimizer. Read through the agreement and accept it. Hit enter to all prompts to finalize the install of Zend Optimizer.

## Create and Install an SSL Certificate

1. Change back into your home directory.

```
> cd
```

2. Create a root certificate.

```
> openssl genrsa -des3 -out server.key 1024
```

Set a password for the key.

3. Remove the passphrase from the key to allow auto-startup of apache.

```
> openssl rsa -in server.key -out server.pem
```

Use the same password in step 2.

4. Generate a Certificate Signing Request

```
> openssl req -new -key server.key -out server.csr
```

- Fill out the values for country, state, city.
- When it asks for organizational unit, leave the field blank
- The PEM passphrase is the password you used originally when you generated a root certificate.
- The Common Name is the domain name or IP the server is on.
- The Challenge Password can be a different password than the PEM passphrase.

5. Generate a Self-Signed Certificate

```
> openssl x509 -req -days 60 -in server.csr -signkey server.key -out server.crt
```

Use the same password you used for the PEM passphrase.

6. Install the Private Key and Certificate.

```
> mkdir /usr/local/apache2/conf/ssl.crt
```

```
> cp server.crt /usr/local/apache2/conf/ssl.crt/server.crt
```

```
> mkdir /usr/local/apache2/conf/ssl.key
```

```
> cp server.key /usr/local/apache2/conf/ssl.key/server.key
```

7. Finalize the configuration for Apache to allow SSL

```
> vi /usr/local/apache2/conf/ssl.conf
```

- Find DocumentRoot and change the path to the DocumentRoot found in the httpd.conf
- Change ServerName to your server's domain name or IP address.

8. Start Apache with SSL

```
> cd /usr/local/apache2/bin
```

```
> ./apachectl stop
```

```
> ./apachectl startssl
```

When it asks for a password, use the password from the PEM phrase.

## Installing osCPro

1. Create a custom directory for the package.

```
> cd
> mkdir cpro
> cd cpro
```

2. Copy and extract the osCpro package into the cpro directory.
3. Copy all the contents of the current directory into your DocumentRoot.

```
> cd catalog
> cp * -R -f /usr/local/apache2/htdocs
```

4. Create the configuration files and set their permissions.

```
> cd /usr/local/apache2/htdocs/includes/
> touch configure.php
> chmod 706 configure.php

> cd /usr/local/apache2/htdocs/admin/includes/
> touch configure.php
> chmod 706 configure.php
```

5. Open a web browser and go to <http://yourserver.com/install>
6. Make sure the Webserver Root Directory is properly set to your DocumentRoot.
7. Make sure the Webserver Catalog Directory is set to only /
8. Webserver Administration Tool Directory should be set to /admin/
9. Use the following database settings

**Database Server:** localhost  
**Username:** osCommerce  
**Password:** *your password for the MySQL username osCommerce*  
**Database:** osCommerce

10. The database should successfully import. On the server configuration screen, check *Enable SSL Connections*. Make sure all other settings and directory paths are correct. Check Store the Sessions in the Database and continue.
11. Fix a bug in osCommerce that attempts to access MySQL with negative limits.

```
> vi /usr/local/apache2/htdocs/admin/includes/classes/split_page_results.php
```

In vi, type colon, slash offset.

```
:/offset
```

Underneath the line that begins with \$offset, insert the following lines:

```
if($offset < 0){
$offset = 0;
}
```

Save and quit from vi. Edit the next file.

```
> vi /usr/local/apache2/htdocs/includes/classes/split_page_results.php
```

In vi, type colon, slash offset.

```
:/offset
```

Underneath the line that begins with \$offset, insert the following lines:

```
if($offset < 0){
$offset = 0;
}
```

Save and quit from vi.

12. Set a cron job to run `subscription_automation.php` once a day.

```
> crontab -e
```

Vi should start up with a blank crontab file. Insert the following line:

```
0 0 * * * php /usr/local/apache2/htdocs/subscription_automation.php
```

Save and quit from vi.

13. Set permissions for osCPro.

```
> chmod 705 /usr/local/apache2/htdocs/includes/configure.php
> chmod 705 /usr/local/apache2/htdocs/admin/includes/configure.php
> chmod 777 /usr/local/apache2/htdocs/images
```

14. Delete the osCPro install directory.

```
> rm -r -f /usr/local/apache2/htdocs/install
```